Università degli Studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Triennale in
Statistica per le Tecnologie e le Scienze

Relazione Finale

Clustering con reti neurali multistrato. Un'analisi di fattibilità su dati astronomici.

Relatore Prof. Alessandra R. Brazzale
Dipartimento di Scienze Statistiche

Laureando Marco Bolzonella
Matricola 2001331

Anno Accademico 2022/2023

Indice

Introduzione

Nel campo dell'astronomia l'evoluzione tecnologica ha portato ad una crescente disponibilità di dati. Questo ha reso fondamentale l'esplorazione di nuove metodologie analitiche per estrarre informazioni significative da dataset vasti e complessi, come nel caso dei cataloghi stellari. A tale proposito, l'obiettivo di questa tesi è esaminare l'efficacia delle reti neurali multistrato come metodologia innovativa per il clustering di stelle variabili, confrontandolo con metodi tradizionali come il k-means e lo hierarchical clustering.
Il clustering di stelle variabili è un'importante area di ricerca nell'astronomia, in quanto aiuta a identificare e raggruppare le stelle che mostrano variazioni di luminosità nel corso del tempo. Queste variazioni di luminosità possono fornire preziose informazioni sulla natura e l'evoluzione delle stelle. Come caso studio è stato selezionato il dataset preso in analisi da Dubath et al., 2011, composto da rilevazioni principalmente provenienti dalla missione Hipparcos (Van Leeuwen, 1997) e riclassificate secondo gli standard dell'International Variable Star Index (Watson et al., 2016).
Nell'ambito di questa relazione, verranno implementati e valutati i tre metodi di clustering utilizzando il linguaggio di programmazione R nella versione 4.3.0. Verranno prese in considerazione come metriche di valutazione l'indice di silhouette e l'indice di Rand aggiustato. I risultati ottenuti dalla comparazione dei metodi di clustering considereranno la qualità del raggruppamento, la separazione dei cluster e la capacità di rilevare correttamente le stelle variabili presenti nel dataset. Attraverso questo studio comparativo, si spera di determinare quale metodo di clustering si adatti meglio alle peculiarità delle stelle variabili.

Capitolo 1

L'insieme di dati

1.1 Stelle variabili

Le stelle variabili sono stelle che mostrano variazioni periodiche o non periodiche nella loro luminosità nel corso del tempo. Studiarle ci permette di calibrare le distanze cosmiche e comprendere l'evoluzione stellare, il che le rende oggetti di grande interesse e ricerca nella cosmologia moderna.
Si possono distinguere in due macro categorie: le intrinseche e le estrinseche. La variabilità delle prime è causata da cambiamenti delle proprietà fisiche della stella, mentre la variabilità delle estrinsiche è causata da proprietà esterne, come dalle eclissi o dalla rotazione. Le stelle variabili intrinseche si possono suddividere in:

- pulsanti. Le pulsazioni stellari sono il risultato di instabilità termica o meccanica all'interno della stella. Queste pulsazioni provocano espansioni e contrazioni periodiche della stella, riflettendosi in variazioni cicliche nella luminosità.

- eruttive. Variano a causa dei bagliori violenti, eruzioni ed espulsioni di massa sui loro strati superficiali.

- cataclismatiche o esplosive. Stelle che subiscono un irregolare e cataclismico cambiamento nella loro luminosità come le supernove, per poi ricadere in uno stato dormiente.

Le stelle variabili estrinseche invece si possono suddividere in:

- binarie ad eclisse. Stelle binarie che, viste dalla terra, occassionalmente si eclissano a vicenda mentre orbitano così da fornire preziose informazioni sulla dinamica e le proprietà delle stelle coinvolte.

- rotanti. La presenza di macchie stellari, analoghe alle macchie solari, su alcune stelle può portare a variazioni nella luminosità durante la loro rotazione. Queste macchie sono aree più fredde e meno luminose sulla superficie stellare, e quando una macchia viene portata in vista, si osserva una temporanea diminuzione nella luminosità.

1.2 Fonti dei Dati

Si è esaminato il dataset preso in analisi da Dubath et al., 2011, composto da rilevazioni principalmente proveninte dalla missione Hipparcos (Van Leeuwen, 1997) (Tab. 1.1) e riclassificate secondo gli standard dell'International Variable Star Index (Watson et al., 2016) .

1.2.1 Catalogo Hipparcos

La missione di astrometria spaziale Hipparcos è stata un progetto dell'Agenzia spaziale europea (ESA) che ha individuato con precisione la posizione di oltre centomila stelle e di oltre un milione di stelle con minor accuratezza. Lanciato nel 1989, è stata la prima missione ad essere dedicata alla misurazione di posizioni, distanze, movimenti, luminosità e colori delle stelle.
Ogni stella finita nel catalogo è stata scansionata circa 100 volte nell'arco di 3 anni e mezzo. Oltre alle stelle misurate nella missione Hipparcos sono state incluse le stelle RS e BY elencate nella terza edizione del catalogo delle stelle binarie cromosfericamente attive (Eker et al., 2008). Gli elenchi delle stelle del tipo GDOR, SPB e BCEP sono stati forniti da comunicazioni personali di P. De Cat e degli LPV da T. Lebzelter. Inoltre, per gli ACV e gli SXARI, sono state incluse le stelle dell'elenco fornito da comunicazione personale di I.I. Romanyuk. Il risultante catalogo Hipparcos, composto da oltre 118.200 stelle, fu pubblicato nel 1997 (Perryman et al., 1997).

1.2.2 VSX-AAVSO

VSX è un catalogo online avviato da un astronomo amatore, Christopher Watson, per un gruppo di astrofili dell'American Association of Variable Star Observers (AAVSO). Nel VSX i dati delle stelle variabili vengono resi disponibili, mantenuti e rivisti per due volte al mese per stare al passo con la letteratura. L'aggiornamento del 13 giugno 2010 è stata adottata da Dubath et al., 2011.
Per questa analisi sono state escluse tutte le stelle del catalogo AAVSO elencate come

Mira, SR, LB o SARV che non erano nell'elenco Lebzelter di LPV.

La tabella 1.1 mostra le categorie di stelle variabili presenti nel dataset con il rispettivo numero di soggetti misurati per tipo di stella. Per maggiori dettagli sulle classi di stelle variabili è possibile consultare la monografia Percy, 2007.

TABELLA 1.1: Lista delle 23 categorie di stelle variabili con rispettiva categoria, referenza principale, acronimo associato e corrispondente numerosità campionaria. L'acronimo p.c sta per comunicazione personale.

Tipo di stella	Categoria	Referenza	Acronimo	Num
Binaria ad eclisse	Binaria ad eclisse	Hipparcos	EA	228
	Binaria ad eclisse	Hipparcos	EB	225
	Binaria ad eclisse	Hipparcos	EW	107
Ellissoidale	Rotante	Hipparcos	ELL	27
Variabili a lungo periodo	Pulsante	Lebzelter(p.c)	LPV	285
RV Tauri	Pulsante	AAVSO	RV	5
W Virginis	Pulsante	AAVSO	CWA	9
	Pulsante	AAVSO	CWB	6
δ Cepheid	Pulsante	AAVSO	DCEP	189
(prima sfumatura)	Pulsante	AAVSO	DCEPS	31
(multimodale)	Pulsante	AAVSO	CEP(B)	11
RR Lyrae	Pulsante	AAVSO	RRAB	72
	Pulsante	AAVSO	RRC	20
γ Doradus	Pulsante	De Cat(p.c)	GDOR	27
δ Scuti	Pulsante	AAVSO	DSCT	47
(bassa amplitudine)	Pulsante	AAVSO	DSCTC	81
β Cephei	Pulsante	De Cat(p.c)	BCEP	30
Pulsa lentamente B	Pulsante	De Cat(p.c)	SPB	81
B linea di emissione	Cataclismatica/esplosiva	AAVSO		
γ Cassiopeiae	Cataclismatica/esplosiva	AAVSO	BE+GCAS	13
α Cygni	Pulsante	AAVSO	ACYG	18
$\alpha - 2$ Canum Venaticorum	Rotante	Romanyuk(p.c)	ACV	77
SX Arietis	Pulsante	Romanyuk(p.c)	SXARI	7
RS Canum Venaticorum and	Rotante	Eker et al. (2008)		
BY Draconis	Rotante	Eker et al. (2008)	RS+BY	35
			Totale	1661

È possibile unire in 16 gruppi le stelle variabili, andando a congiungere categorie di stelle più simili tra loro (Tab. 1.2).
Si può notare dalle distribuzioni di frequenza che le classi presenti nei dati sono fortemente sbilanciate. Ad esempio sono state misurate 590 stelle binarie ad ellisse e soltanto 27 elissoidali.

TABELLA 1.2: Lista delle 16 famiglie di stelle variabili con il loro acronimo associato e rispettiva numerosità campionaria.

Famiglia di stelle	Categoria	Acronimo	Num
α Cygni	Pulsante	ACYG	18
$\alpha - 2$ Canum Venaticorum	Rotante	ACV	77
β Cephei	Pulsante	BCEP	30
B linea di emissione	Cataclismatica o esplosiva		
γ Cassiopeiae	Cataclismatica o esplosiva	BE+GCAS	13
δ Cepheid	Pulsante	DCEP+DCEPS+CEP(B)	231
W Virginis	Pulsante	CWA+CWB	15
δ Scuti	Pulsante	DSCT+DSCTC	128
Binarie ad eclisse	Binarie ad eclisse	EA+EB+EW	590
Ellissoidali	Rotante	ELL	27
γ Doradus	Pulsante	GDOR	27
Variabili a lungo periodo	Pulsante	LPV	285
RR Lyrae	Pulsante	RRAB+RRC	92
RS Canum Venaticorum and	Rotante		
BY Draconis	Rotante	RS+BY	35
RV Tauri	Pulsante	RV	5
Pulsa lentamente B	Pulsante	SPB	81
SX Arietis	Pulsante	SXARI	7
		Totale	1661

1.3 Attributi

Nel insieme di dati considerato sono presenti 45 attributi, ciascuno dettagliato nella tabella 1.3. Questi attributi coprono una vasta gamma di informazioni, dalle proprietà stellari come il colore medio e la luminosità assoluta, alle caratteristiche delle curve di luce, come il periodo e l'ampiezza. Alcuni attributi si concentrano anche sulla forma delle curve di luce ripiegate. In sintesi, questi 45 attributi possono essere organizzati in cinque categorie distinte in base alle informazioni che raccolgono, come indicato nella tabella 1.4.

TABELLA 1.3: Descrizione dei 45 attributi presenti nel dataset originale.

Attributo	Descrizione
hip	Identificatore Hipparcos
cat	P = periodica, X = "irrisolte"
type	Vera classe di appartenenza della stella
p2pScatterOnDetrendedTS	Dispersione punto per punto della serie temporale della luminosità dopo aver rimosso un trend polinomiale
p2pScatterOnFoldedTS	Dispersione punto per punto della serie temporale della luminosità dopo aver trovato un periodo e un ripiegamento di fase
scatterOnResidualTS	Radice quadrata della varianza dei residui dopo la modellazione
Raw_WeightedStdDev	Deviazione standard ponderata della luminosità
Raw_WeightedSkewness	Asimmetria ponderata della luminosità
Raw_WeightedKurtosis	Curtosi ponderata della luminosità
Raw_PercentileRange10	0,1 quantile meno la mediana delle luminosità grezze
stetsonJ	Misure di correlazione tra valori di luminosità ravvicinati
stetsonJweighted	Misure di correlazione tra valori di luminosità ravvicinati
stetsonK	Misure di correlazione tra valori di luminosità ravvicinati
WstetsonJ	Misure di correlazione tra valori di luminosità ravvicinati
WstetsonJweighted	Misure di correlazione tra valori di luminosità ravvicinati
WstetsonK	Misure di correlazione tra valori di luminosità ravvicinati
logPnonQso	Misura della variabilità stocastica, componenti delle curve di luce
logPqso	Misura della variabilità stocastica, componenti delle curve di luce
qsoVar	Misura della variabilità stocastica, componenti delle curve di luce
nonQsoVar	Misura della variabilità stocastica, componenti delle curve di luce
LogPeriod	Logaritmo in base 10 del periodo in giorni
LogAmplitude	Algoritmo in base 10 dell'amplitufine picco a picco (dall'adattamento del modello armonico e dalla ricostruzione della curva di luce adattata)
HarmNum	Il più alto ordine significativo di termini armonici nei minimi quadrati
A11	Amplitudine del primo termine armonico
A12	Amplitudine del secondo termine armonico
PH12	Fase relativa del secondo termine armonico
A13	Amplitudine del terzo termine armonico
PH13	Fase relativa del terzo termine armonico
A14	Amplitudine del quarto termine armonico
PH14	Fase relativa del quarto termine armonico
A15	Amplitudine del quinto termine armonico
PH15	Fase relativa del quinto termine armonico
logA11minus	$\log_{10}(1 + abs(A11 - \sqrt{(\sum A1j^2)}))$
logA12_A11	$\log_{10}(1 + A12/A11)$
logA13_A12	$\log_{10}(1 + A13/A12)$
absGlat	Valore assoluto della latitudine galattica
Glat	Latitudine galattica
Glon	Longitudine galattica
Parallax	La parallasse dell'oggetto (in milliarcosec) equivalente alla distanza
Absolute_Mag00	Stima della luminosità assoluta dell'oggetto
BV_Color	Colori calcolati con luminosità visiva. B può essere considerato "blu", V è "visivo", I è un filtro rosso vicino all'infrarosso
VI_Color	Colori calcolati con luminosità visiva. B può essere considerato "blu", V è "visivo", I è un filtro rosso vicino all'infrarosso
JmK	J, H, K sono tutti filtri infrarossi
JmH	J, H, K sono tutti filtri infrarossi
HmK	J, H, K sono tutti filtri infrarossi

TABELLA 1.4: Attributi presenti nel dataset originale raccolti per funzionalità.

Tipo di attributo	Attributo
Attributi che riassumono i miglioramenti ottenuti dopo alcuni passaggi di modellazione	p2pScatterOnDetrendedTS p2pScatterOnFoldedTS scatterOnResidualTS
Attributi che riassumono la distribuzione delle magnitudini osservate	Raw_WeightedStdDev Raw_WeightedSkewness Raw_WeightedKurtosis Raw_PercentileRange10 stetsonJ stetsonJweighted stetsonK WstetsonJ WstetsonJweighted WstetsonK
Attributi che quantificano la forza della variabilità stocastica	logPnonQso logPqso qsoVar nonQsoVar
Attributi relativi alla ricerca del periodo e alla modellazione armonica	LogPeriod LogAmplitude HarmNum A11, A12, A13, A14, A15 PH12, PH13, PH14, PH15 logA11minus
Attributi relativi all'astrofisica	absGlat Glat Glon Parallax Absolute_Mag00 BV_Color VI_Color JmK, JmH, HmK

Capitolo 2

Metodologia di analisi

In questo capitolo verranno esaminate le strategie e gli algoritmi adottati in questa tesi. Per esplorare quali tipi di dati funzionano meglio con gli algoritmi di clustering si è effettuata l'analisi delle componenti principali. Per lo stesso motivo si è deciso di normalizzare i dati tramite la normalizzazione del simplesso sui dati di partenza.
Per ottimizzare il costo computazionale, è stato deciso di ridurre le variabili misurate nel dataset da 43 a 12. Questo è stato realizzato attraverso l'impiego di un algoritmo di selezione basato sul random forest (algoritmo di machine learning che si basa sulla creazione di un "bosco" (forest) di alberi decisionali (Parmar et al., 2019)), che ha valutato l'importanza di ciascuna variabile nella capacità del modello di effettuare previsioni precise sul clustering.
Infine, per effettuare l'analisi dei cluster, si sono adottati gli algoritmi K-means, h-clust e le reti neurali multistrato. I risultati ottenuti con ciascun algoritmo sono stati misurati tramite l'indice della silhouette.

2.1 Riduzione della dimensione

2.1.1 Analisi delle componenti principali

L'analisi delle componenti principali (PCA) è uno strumento per la riduzione dimensionale dei dati. L'obiettivo della PCA è quello di proiettare un insieme complesso di dati in uno spazio a dimensione inferiore, preservando al massimo la varianza dei dati originali (Kurita, 2019).

Il cuore della PCA risiede nella trasformazione dei dati originali in un nuovo sistema di coordinate, definito da un insieme di assi chiamati "componenti principali". Una componente principale è una direzione nel dataset originale lungo la quale i dati variano

di più. Ogni componente principale è una combinazione lineare delle variabili originali nel dataset. La prima componente principale è la direzione nel dataset in cui la varianza dei dati è massima e le componenti successive catturano via via meno varianza, ma sono ortogonali alle componenti precedenti, garantendo così l'indipendenza lineare tra le nuove variabili.

L'utilità della PCA è multiforme: la riduzione dimensionale facilita la visualizzazione dei dati in uno spazio a due o tre dimensioni, rendendo più agevole l'interpretazione dei pattern sottostanti. Inoltre, la PCA è ampiamente utilizzata per eliminare la multicollinearità nei dati, semplificando così l'analisi statistica.
L'analisi delle componenti principali è essenziale nella preparazione dei dati per algoritmi di machine learning, dove la riduzione dimensionale può migliorare l'efficienza e la precisione dei modelli.

2.1.2 Normalizzazione di Dirichlet

La normalizzazione è un processo utilizzato nell'analisi dei dati per trasformare le variabili in modo che abbiano una scala comune o uno specifico range, così da poter confrontare le variabili nonostante le unità di misura differenti.
Una delle tecniche di normalizzazione più utilizzate è la normalizzazione tramite il simplesso, anche nota come Dirichlet normalization (Aitchison, 1985). Questa è basata sulla distribuzione di Dirichlet, che è una distribuzione di probabilità continua utilizzata per modellare proporzioni.

Nello spazio euclideo, un simplesso è il più semplice poliedro convesso possibile. Per un certo numero n di dimensioni, è costituito da $n+1$ vertici, n spigoli e n facce. Nel contesto della distribuzione di probabilità e dell'ottimizzazione, il termine "simplesso" è spesso utilizzato per riferirsi allo spazio convesso delle probabilità valide, in cui ogni coordinata è non negativa e la somma di tutte le coordinate è uguale a 1.

La normalizzazione tramite il simplesso segue la funzione di ripartizione (CDF) per convertire le variabili casuali non negative in modo che la loro somma sia uguale a 1. Per esempio, si supponga di avere n variabili casuali, ciascuna con la propria distribuzione di probabilità continua. La normalizzazione del simplesso può essere ottenuta calcolando la CDF di ciascuna variabile, ovvero $F_i(x)$ per $i = 1, 2, \ldots, n$, dove x è il valore che la variabile aleatoria X_i può assumere.
Per normalizzare si è seguito il seguente procedimento:

1. è stata calcolata la funzione di ripartizione di ciascuna variabile.
   $F_i(x) = P(X_i \leq x)$

2. si è calcolata la somma cumulativa delle CDF .
   $S(x) = F_1(x) + F_2(x) + \ldots + F_n(x)$

3. sono state normalizzate le variabili casuali dividendole per la somma cumulativa.
   $Y_i = F_i(x)/S(x)$

Dove $Y_1, Y_2, \ldots, Y_n$ sono le variabili normalizzate e soddisfano $Y_1 + Y_2 + \ldots + Y_n = 1$, che rappresenta il vincolo del simplesso. In questo modo, le variabili preservano i legami di dipendenza tra di loro attraverso l'utilizzo delle rispettive funzioni di ripartizione.

L'approccio del simplesso è cruciale in scenari in cui è necessario confrontare o analizzare proporzioni tra diverse categorie o gruppi. Normalizzando i dati attraverso la distribuzione di Dirichlet, si garantisce che le proporzioni rimangano coerenti e confrontabili, evitando distorsioni dovute a scale diverse o varianze impreviste nei dati.

2.1.3 Random Forest

Il random forest è un'algoritmo che opera attraverso l'assemblaggio di numerosi alberi decisionali, ognuno addestrato su un sottoinsieme casuale del dataset (Parmar et al., 2019).
Questa diversificazione nella costruzione degli alberi e l'introduzione di casualità nel processo di apprendimento mitigano l'overfitting, consentendo all'algoritmo di catturare pattern complessi nei dati senza sacrificare la capacità di generalizzazione. Infatti, l'overfitting è un problema nel machine learning che si verifica quando un modello è allenato troppo accuratamente sui dati di addestramento e perde la sua capacità di generalizzare su nuovi dati.
Ogni albero nel random forest contribuisce al processo decisionale attraverso votazione, e il risultato finale è una predizione ponderata, che ne riflette il contributo.

Viene definito dunque un algoritmo che misura quanto ciascuna variabile contribuisca alla riduzione dell'errore nelle previsioni del modello e confronta le previsioni fatte da ciascun albero nell'ensemble. Per farlo si deve introdurre il termine OOB (out-of-bag), ovvero le osservazioni che non sono state utilizzate per addestrare un albero specifico. Poiché ciascun albero nel random forest è addestrato su un diverso sottoinsieme di dati, i dati che non sono stati utilizzati per l'addestramento di un particolare albero possono essere utilizzati per valutare le prestazioni di quell'albero specifico.
Per ogni variabile nel dataset, l'algoritmo di selezione procede seguendo quattro step:

1. Vengono mescolati i valori di una variabile nel dataset.

2. Si valida il modello (random forest) con le osservazioni OOB.

3. Si stima un'errore di previsione per le osservazioni OOB.

4. Si trova la differenza tra l'errore ottenuto al punto 3 e l'errore ottenuto senza aver mescolato i valori della variabile.

Così si ottiene quanto la variabile in esame influenza la capacità del modello di fare previsioni accurate sul clustering, e dunque la sua importanza (Genuer et al., 2010).

2.2 Clustering e apprendimento non supervisionato

È fondamentale comprendere la differenza tra apprendimento supervisionato e non supervisionato (Berry et al., 2019). Il primo è un approccio in cui il modello viene addestrato su dati etichettati, ovvero dati in cui le categorie o le classi sono conosciute in anticipo. Nel caso in analisi, corrisponderebbe aconoscere il tipo di stella variabile.
L'apprendimento non supervisionato si basa invece su dati non etichettati, tra i quali il modello cerca autonomamente di identificare pattern o strutture, come cluster, senza alcuna supervisione umana diretta. Quest'ultimo è particolarmente utile quando non si hanno informazioni pregresse sulle categorie o quando si vogliono esplorare nuove relazioni nei dati.

Il clustering è una tecnica di analisi non supervisionata che mira a partizionare un insieme di dati in gruppi omogenei, in modo che gli elementi all'interno di ciascun gruppo siano simili tra loro e diversi dagli elementi negli altri gruppi (Madhulatha, 2012). Di seguito verrano descritte le tecniche di clustering utilizzate in questa tesi, a partire da K-means, h-clust e per concludere le reti neurali multistrato, al fine di capire se le reti neurali possano rivelarsi un metodo valido per effettuare clustering sulle stelle variabili.

2.2.1 K-means

L'algoritmo K-means, a differenza di altri metodi, permette di dividere i dati in un numero predefinito di cluster. Questo è basato su un approccio iterativo che cerca di minimizzare la somma dei quadrati delle distanze tra le osservazioni e i centroidi dei cluster a cui appartengono. Un centroide è un punto che rappresenta il centro geometrico di un insieme di punti in uno spazio multidimensionale.
In altre parole, la funzione obiettivo del K-means è definita come:

$$J = \sum_{i=1}^{n} \sum_{j=1}^{k} ||x_i - c_j||^2 \tag{2.1}$$

dove J è la funzione obiettivo da minimizzare, n è il numero di di osservazioni nel dataset, k è il numero di cluster, x_i è l'osservazione $i$esima, c_j è il centroide del $j$esimo cluster e $||x_i - c_j||^2$ rappresenta la distanza euclidea tra il punto x_i e il centroide c_j.
Questo algoritmo procede attraverso una serie di passaggi chiave:

1. Vengono selezionati casualmente k punti dal dataset come centroidi iniziali. Questi punti rappresentano i centri iniziali dei cluster che si sta cercando di creare.

2. Per ogni osservazione nel dataset, l'algoritmo calcola la sua distanza rispetto a ciascun centroide. Il punto viene quindi assegnato al cluster del centroide più vicino, basandosi sulla distanza calcolata.

3. Una volta che tutti i punti sono stati assegnati ai cluster, l'algoritmo ricalcola i centroidi per ogni cluster prendendo la media di tutti i punti assegnati ai cluster. Ora i centroidi rappresentano i nuovi centri dei rispettivi cluster.

4. I passaggi 2 e 3 vengono ripetuti iterativamente fino a quando i centroidi non cambiano significativamente tra le iterazioni o un numero predefinito di iterazioni è stato raggiunto. A questo punto, i cluster risultanti sono considerati stabili e l'algoritmo restituisce i cluster finali.

Questo algoritmo è sensibile alla scelta iniziale dei centroidi e può convergere verso minimi locali della funzione obiettivo. Pertanto, spesso è utile eseguirlo più volte con diverse inizializzazioni dei centroidi per ottenere una soluzione ottimale o quasi-ottimale.

2.2.2 H-clust

Il metodo di clustering gerarchico h-clust, anche noto come clustering gerarchico agglomerativo, è un algoritmo che inizialmente considera ogni oggetto come un singolo cluster e successivamente combina iterativamente i cluster più simili fino a ottenerne uno unico che contiene tutti gli oggetti.
L'algoritmo h-clust opera in base a una misura di similarità o dissimilarità tra gli oggetti che viene definita in base alle caratteristiche degli oggetti in analisi. Nel caso dei vettori numerici, la distanza euclidea è la più indicata come misura di similarità data la sua sensibilità ai cambiamenti nelle coordinate dei punti e al fatto che tiene conto delle differenze nelle dimensioni dei vettori. Ovvero, se un'unità in una dimensione è uguale a un'unità in un'altra dimensione, la loro distanza sarà la stessa indipendentemente dalla scala assoluta delle dimensioni. La distanza euclidea è una misura di distanza

tra due punti in uno spazio $n-$dimensionale e, tra due punti P e Q con coordinate $(x_1, x_2, \ldots, x_n)$ e $(y_1, y_2, \ldots, y_n)$, è data da

$$D_e = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}. \tag{2.2}$$

Altre misure di similarità sono la distanza di Manhattan e la distanza di Lagrange. La prima misura la distanza come la somma delle differenze assolute tra le loro coordinate, mentre la seconda calcola la distanza tra due punti come la massima differenza tra le loro coordinate lungo le diverse dimensioni.

Infine, il clustering gerarchico agglomerativo offre anche diverse metriche di collegamento, come il collegamento singolo, medio, completo e di Ward.
Il collegamento singolo definisce la distanza tra due cluster come la distanza minima tra i punti di un cluster e quelli dell'altro. Questo significa che due cluster verranno collegati se hanno almeno un punto molto vicino tra loro.
Il collegamento medio definisce la distanza tra due cluster come la media delle distanze tra tutti i punti dei due cluster. Questo tende a produrre cluster più compatti rispetto al collegamento singolo.
Il collegamento completo definisce la distanza tra due cluster come la distanza massima tra i punti dei due cluster. Questo può produrre cluster più coesi, ma la loro costruzione può essere influenzata da outlier.
Infine, il collegamento di Ward cerca di minimizzare la varianza intra-cluster. Durante il processo di fusione, si calcola come la differenza tra la varianza totale dopo la fusione e la somma delle varianze dei cluster originali. Questo approccio tende a produrre cluster compatti e bilanciati.

L'output del metodo h-clust è rappresentato da un dendrogramma, un diagramma ad albero che visualizza la gerarchia di cluster. Ogni oggetto o cluster è rappresentato da una linea verticale sull'asse orizzontale. Queste collegano i cluster in base alla loro similarità, con le linee più lunghe che rappresentano unioni di cluster più distanti e le linee più corte indicanti unioni di cluster vicini. Altezza e posizione delle linee nel dendrogramma indicano la dissimilarità tra i cluster. Un livello di taglio può essere stabilito per determinare il numero di cluster (nella Fig. 2.1 la linea orizzontale in rosso).
L'analisi visiva del dendrogramma aiuta a comprendere le relazioni tra i gruppi e a prendere decisioni sul numero ottimale di cluster.

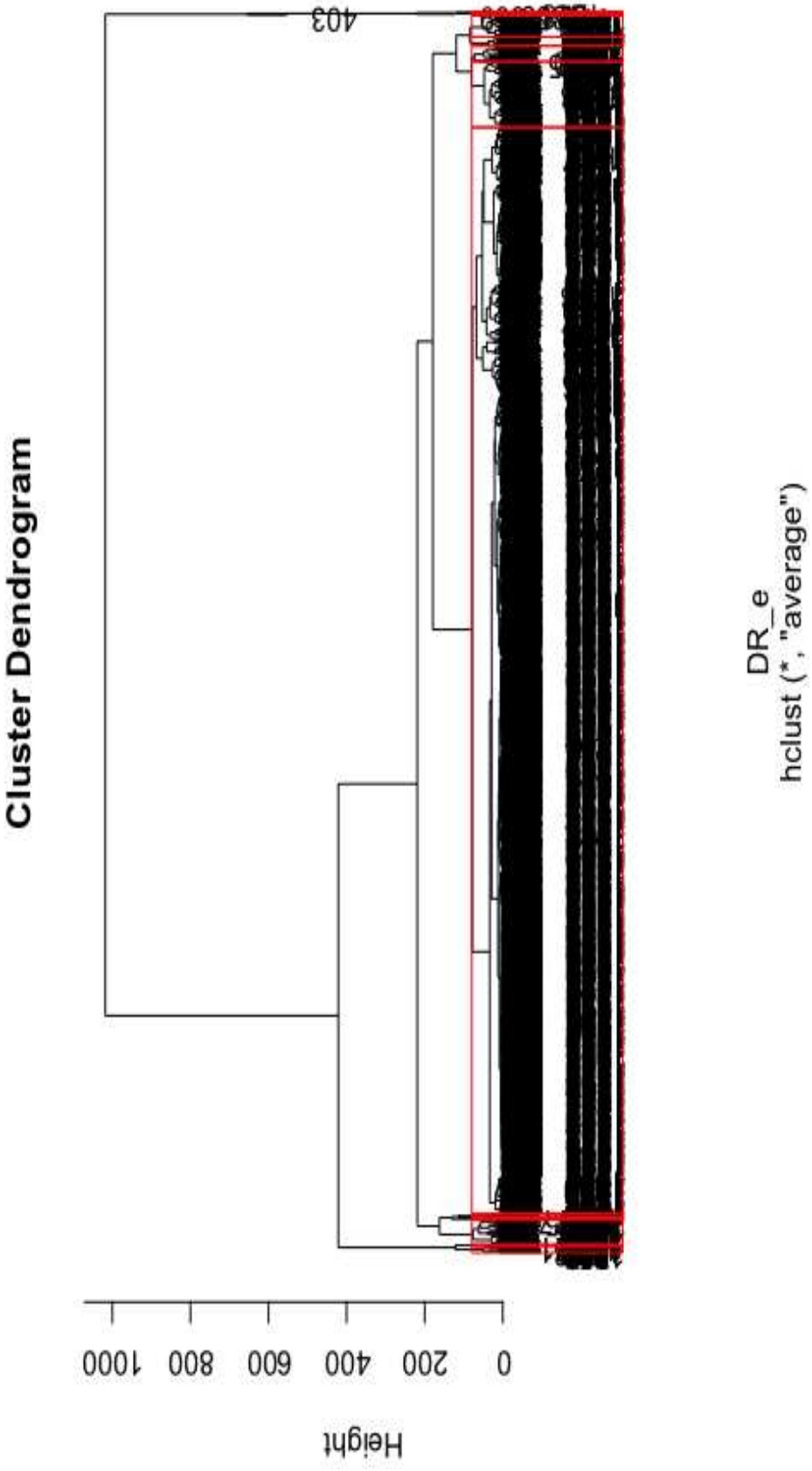


FIGURA 2.1: Esempio di dendogramma tratto dall'analisi del dataset con il collegamento medio.

2.3 Deep Learning: Reti neurali multistrato

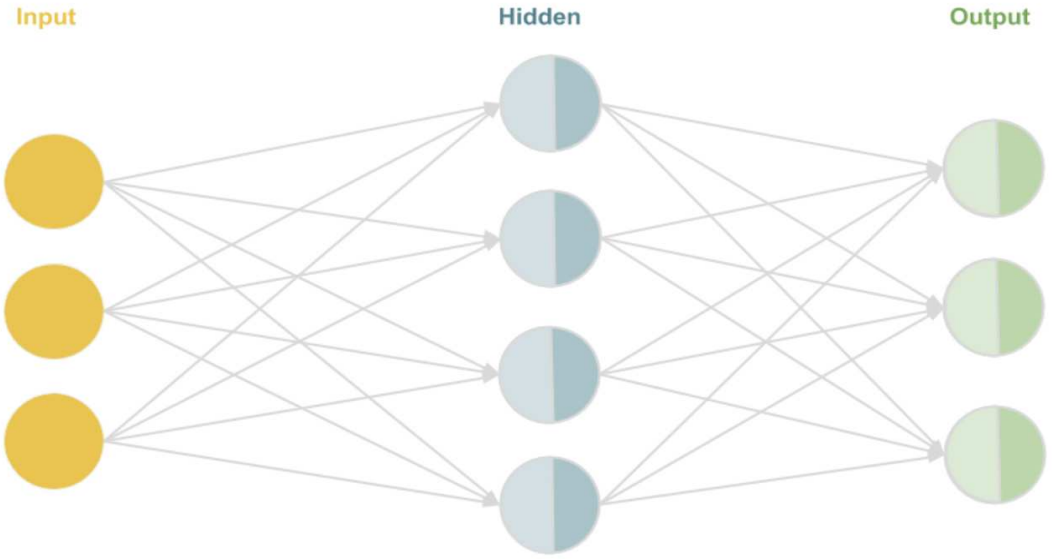
Il deep learning è un sotto campo del machine learning basato sulle reti neurali. Questa metodologia nasce agli inizi degli anni settanta, oggi tornata in voga a causa della crescente disponibilità di dati, di nuovi hardware in grado di effettuare in tempi ridotti la stima di milioni di parametri e grazie a nuove strategie di ottimizzazione e regolarizzazione.

Le reti neurali si ispirano al funzionamento del cervello umano e sono potenti strumenti nell'ambito dell'analisi dei dati e dell'apprendimento automatico. I neuroni artificiali, che sono l'unita computazionale di base per una rete neurale, sono organizzati in strati, ognuno dei quali ha un ruolo specifico nel processo di apprendimento.
Il primo strato, chiamato strato di input, riceve i dati in ingresso. Se x è un vettore p dimensionale, allora si avranno p neuroni nel primo strato.
Successivamente, i dati vengono elaborati attraverso uno o più strati intermedi, chiamati strati nascosti, che apprendono rappresentazioni complesse dei dati. Nascosti poichè non si è al corrente dei loro output.
Infine, nello strato di output, la rete restituisce i risultati predetti o le classificazioni desiderate. Ogni neurone nello strato di output corrisponderà ad una classe specifica nel caso della classificazione.
Dato che l'output di uno strato serve da input per lo strato successivo, si dice che le reti neurali multistrato hanno una struttura gerarchica.
Nella Fig. 2.2 si può vedere che il vettore degli input ha tre dimensioni, è inoltre rappresentato uno strato nascosto composto da quattro neuroni e lo strato dei neuroni che forniscono l'output che ci indica che sono presenti tre classi.

All'interno di un neurone gli input vengono ponderati da specifici pesi sinaptici. Ogni connessione tra un input e neurone ha un peso associato. La somma ponderata degli input viene calcolata come $\sum_{i=1}^{n}(x_i \cdot w_i)$ dove x_i è l'input e w_i è il peso associato all'input i esimo.
Dopo aver calcolato la somma ponderata degli input, questa viene passata attraverso una funzione di attivazione che determina se il neurone deve essere attivato o meno. Una funzione di attivazione comune è la funzione sigmoide che restituisce valori nell'intervallo tra 0 e 1, ma ci sono molte altre funzioni di attivazione utilizzate nelle reti neurali, come la funzione ReLU (Rectified Linear Unit) che restituisce 0 per input negativi e l'input stesso per input positivi.

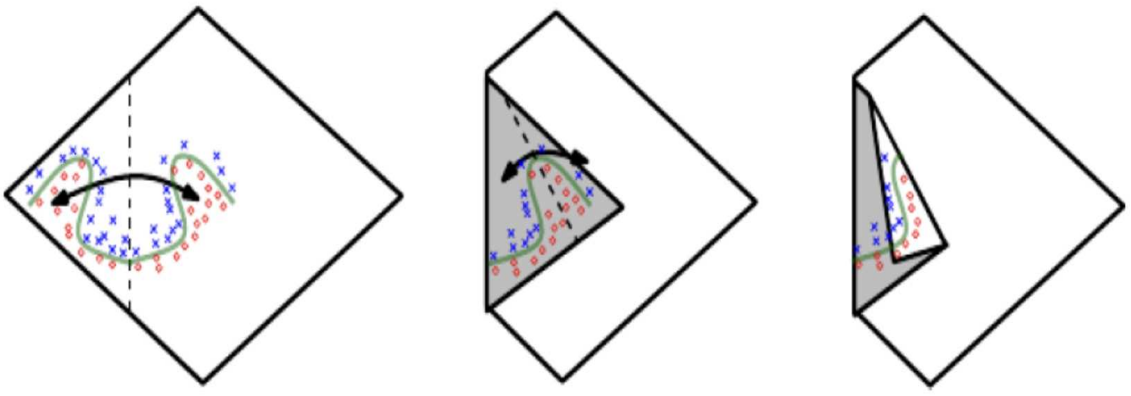
Una rete neurale composta da un singolo neurone sarà in grado di imparare solo modelli di classificazione lineari, ma reti neurali più complesse riescono a modificare

FIGURA 2.2: Rappresentazione di una rete neurale multistrato.



lo spazio delle covariate semplificandolo, così da compiere classificazione non lineare. Maggiore sarà il numero di strati, più la rete viene detta profonda e più si riuscirà ad ottenere rappresentazioni astratte. Nella figura 2.3 è rappresentato graficamente come gli strati delle reti neurali modificano lo spazio delle variabili durante un compito di classificazione binaria. Inizialmente, i dati presentano una struttura non lineare. La rete neurale presa in esempio, composta da tre strati, riconfigura lo spazio delle variabili per ottenere una rappresentazione più chiara, semplificando così il compito della classificazione.

FIGURA 2.3: Un'intuitiva rappresentazione geometrica del vantaggio delle reti neurali (Montufar et al., 2014).



Aumentare il numero di strati può specializzare la rete neurale solo nei dati su cui viene addestrata al punto da adattarsi anche al rumore o alle variazioni casuali presenti. Questo fenomeno viene chiamato overfitting e può essere provocato da cause differenti:

- Il dataset di addestramento è piccolo, la rete neurale potrebbe memorizzare i dati invece di imparare da essi.
- L'eccessiva complessita della rete rispetto ai dati, così che anzichè riconoscere i pattern all'interno del dataset, memorizza i dati stessi.
- Un numero eccessivo di epoche per l'addestramento della rete.
- La presenza di rumore o variazioni casuali. La pulizia e la normalizzazione dei dati possono aiutare a ridurre questo problema.
- Un forte sbilanciamento delle classi.

2.3.1 Clustering con le reti neurali

Negli ultimi anni, le reti neurali sono emerse come una potente alternativa ad algoritmi di clustering più tradizionali come K-means e h-clust. L'utilizzo delle reti neurali per il clustering, noto come clustering basato su autoencoder (Song et al., 2014), offre una prospettiva innovativa per l'analisi dei dati complessi.

Le reti neurali, in particolare gli autoencoder, hanno dimostrato una straordinaria capacità di apprendere rappresentazioni latenti dei dati. Nella fase di addestramento, un autoencoder apprende una rappresentazione compressa dei dati di input, catturando le caratteristiche più rilevanti e significative. Questa rappresentazione latente può essere sfruttata per eseguire il clustering dei dati in maniera più precisa ed efficiente. I dati, proiettati in uno spazio di rappresentazione appreso dalla rete, possono essere facilmente clusterizzati utilizzando algoritmi di clustering tradizionali.

Un autoencoder è composto da due parti principali: l'encoder e il decoder. L'encoder è la parte della rete neurale che comprime l'input in una rappresentazione di spazio nascosto (rappresentazione latente) di dimensione inferiore rispetto all'input. Questa rappresentazione latente contiene le caratteristiche più rilevanti dell'input mentre elimina il rumore e i dettagli superflui.
Il decoder è responsabile di ricostruire l'input originale a partire dalla rappresentazione latente prodotta dall'encoder. L'obiettivo del decoder è produrre un'uscita che sia il più simile possibile all'input originale. In altre parole, il decoder cerca di decomprimere la rappresentazione latente per ricostruire i dati di input originali.

L'autoencoder viene addestrato minimizzando l'errore di ricostruzione tra l'input originale e l'output del decoder.

2.4 Metriche di valutazione

La necessità di valutare l'efficacia dei cluster individuati è cruciale per comprendere la struttura sottostante dei dati stessi. Come metriche per misurare la qualità del clustering si è deciso di adottare l'indice di silhouette (Dudek, 2020), l'indice di Rand (Rand, 1971) e la sua versione aggiustata che tiene conto delle somiglianze casuali che potrebbero verificarsi tra i cluster osservati e quelli previsti dagli algoritmi (Hubert and Arabie, 1985).

2.4.1 Silhouette

L'indice di silhouette si presenta come una misura fondamentale in questo ambito, offrendo un'analisi dettagliata ed esaustiva della coerenza e della separazione tra i cluster identificati. Esso misura quanto un punto sia simile agli altri punti all'interno dello stesso cluster (coesione) rispetto agli altri cluster (separazione). Il valore dell'indice può variare da -1 a 1. Un valore vicino a 1 indica che l'osservazione è stata assegnata al cluster corretto, mentre un valore vicino a -1 suggerisce che il cluster a cui è stata assegnata l'osservazione non è appropriato. Un valore vicino a 0 indica che l'osservazione è sul bordo tra due cluster.
La distanza dell'osservazione x_{i*} dal gruppo R_g con n_g osservazioni, di cui x_i è un'elemento, è valutata come

$$D(x_{i*}, R_g) = \frac{1}{n_g} \sum_{x_i \in R_g} d(x_{i*}, x_i). \tag{2.3}$$

Sia R_{g*} il gruppo in cui è inclusa l'osservazione x_{i*} e sia D_0 la distanza di x_{i*} dal gruppo più vicino diverso da quello a cui appartiene

$$D_0 = \min_{g \neq g^*} D(x_{i*}, R_g). \tag{2.4}$$

L'indice di silhouette confronta D_0 con la distanza dal gruppo a cui x_{i*} appartiene mediante

$$S(x_{i*}) = \frac{D_0 - D(x_{i*}, R_{g*})}{\max(D_0, D(x_{i*}, R_{g*}))} \tag{2.5}$$

ed è tanto più grande quanto più x_{i*} è vicino al suo gruppo e distante dagli altri.

2.4.2 Indice di Rand

L'Indice di Rand (RI) è una metrica utilizzata per quantificare la similitudine tra due partizioni di un insieme di elementi. Nel contesto dei problemi di clustering, questa metrica valuta l'accuratezza con cui un algoritmo assegna gli elementi ai gruppi appropriati confrontando le previsioni con le osservazioni, fornendo così una misura della coerenza delle partizioni ottenute. L'Indice di Rand può variare da 0 a 1, dove 1 indica una perfetta concordanza tra le partizioni .
Per calcolarlo, si considerano quattro quantità:

- a: il numero di coppie di elementi che sono nella stessa parte in entrambe le partizioni
- b: il numero di coppie di elementi che sono in parti diverse in entrambe le partizioni.
- c: il numero di coppie di elementi che sono nella stessa parte nella prima partizione, ma in parti diverse nella seconda partizione.
- d: il numero di coppie di elementi che sono in parti diverse nella prima partizione, ma nella stessa parte nella seconda partizione.

Da queste, si ottiene l'indice di Rand attraverso

$$RI = \frac{a+b}{a+b+c+d} \tag{2.6}$$

Quando si tratta di valutare l'efficacia degli algoritmi di clustering su dati non bilanciati, si prende in considerazione l'indice di Rand aggiustato (ARI). Questo tiene in considerazione le somiglianze casuali che potrebbero verificarsi tra i cluster osservati e quelli previsti dagli algoritmi. L'ARI varia da -1 a 1, dove 1 indica una perfetta concordanza tra le partizioni, 0 indica una concordanza casuale e valori negativi indicano una concordanza peggiore di quella casuale.
L'indice di Rand aggiustato si calcola per mezzo di

$$ARI = \frac{RI - Esp[RI]}{\max(RI) - Esp[RI]} \tag{2.7}$$

Dove RI è l'Indice di Rand e Esp[RI] è l'aspettativa dell'Indice di Rand calcolata sotto l'ipotesi di distribuzione casuale delle partizioni.

Capitolo 3

Analisi dei cluster

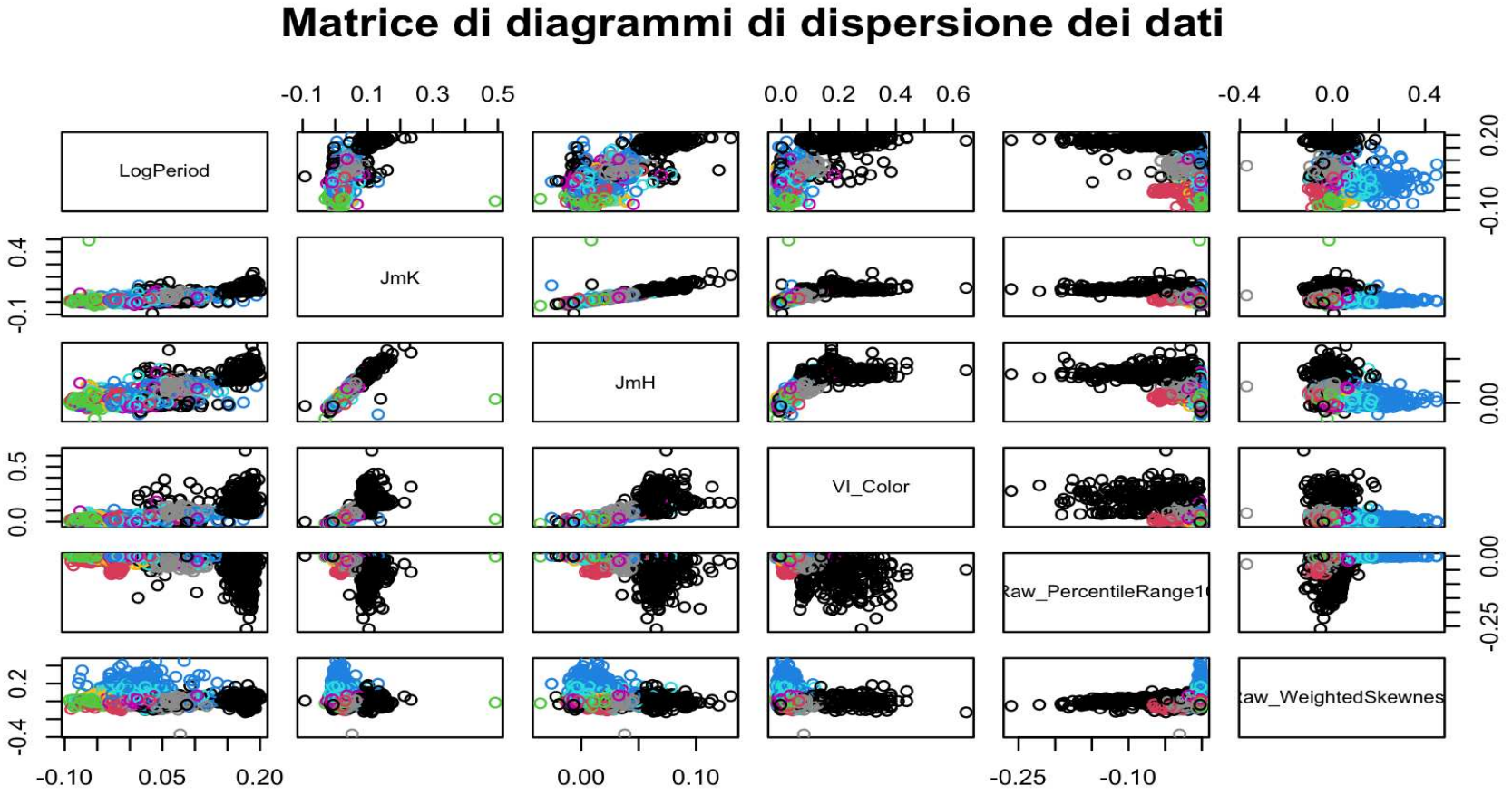
L'obiettivo di questa analisi è se l'utilizzo delle reti neurali multistrato può essere una metodologia innovativa per il clustering di stelle variabili. Per comprenderlo si confronteranno i risultati con metodi tradizionali come il k-means e l'hierarchical clustering.

3.1 Preparazione dei dati

Prima di procedere con l'analisi dei cluster si è deciso di ridurre la dimensionalità del dataset al fine di ottimizzare il costo computazionale degli algoritmi. Per farlo, si è adottato il metodo di selezione delle variabili basato sul Random Forest. Dalle 43 variabili del dataset originale se ne sono selezionate 13 (Tab. 3.1). Da questa selezione emerge che le variabili più rilevanti rappresentano l'ampiezza della frequenza della luce emessa dalla stella, il colore della stella (e quindi la sua temperatura), la periodicità con la quale la luminosità oscilla nel tempo e la frequenza infrarossa in cui è rilevabile. Nelle figure 3.1 e 3.2 si possono osservare le matrici dei diagrammi di dispersione dei dati rispetto le variabili selezionate dall'algoritmo con le osservazioni colorate in base al tipo di stella variabile.

In secondo luogo, si è deciso di normalizzare i dati tramite la normalizzazione di Dirichlet al fine di evitare distorsioni dovute a scale diverse o varianze impreviste nei dati. Inoltre, molti algoritmi di clustering beneficiano dei dati normalizzati in quanto questa procedura assicura che ogni variabile contribuisca in modo equo all'analisi, si riduce l'impatto di variabili con scale molto diverse sulla distanza euclidea e può aiutare gli algoritmi a convergere più rapidamente.

FIGURA 3.1: Diagrammi di dispersione dei dati sulle prime sei variabili selezionate dall'algoritmo. Osservazioni colorate rispetto al tipo di stella variabile.



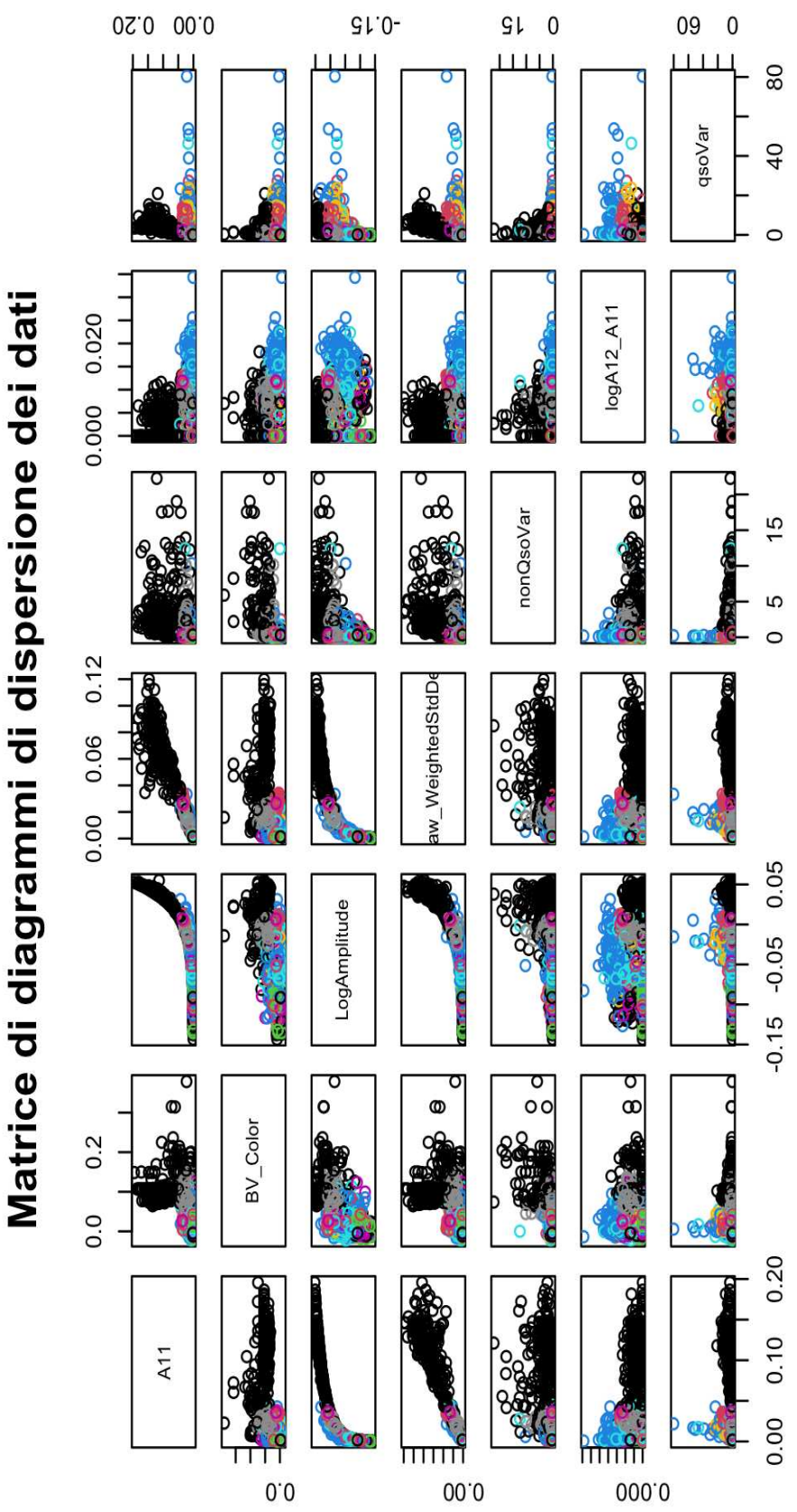
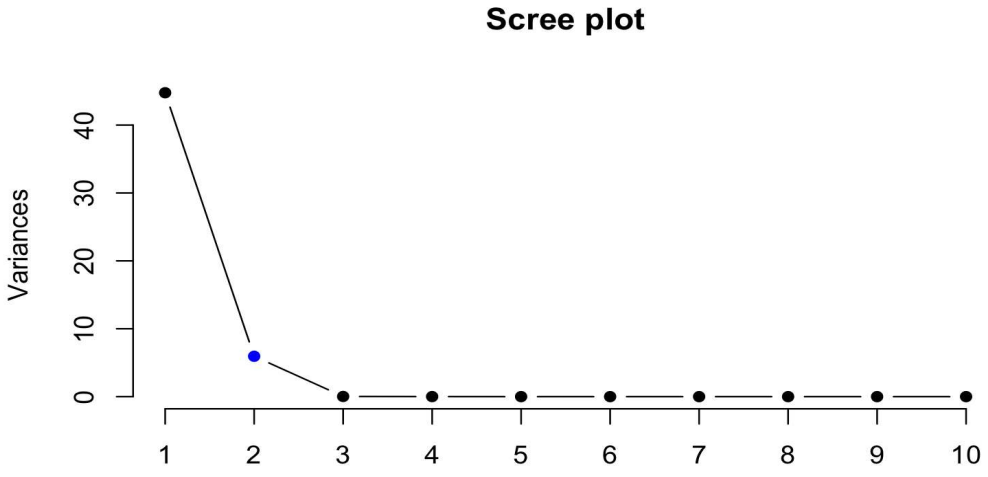


FIGURA 3.2: Diagrammi di dispersione dei dati sulle rimanenti variabili selezionate dall'algoritmo. Osservazioni colorate rispetto al tipo di stella variabile.

Infine, si è svolta l'analisi delle componenti principali con lo scopo di ridurre ulteriormente la dimensionalità dei dati. Per scegliere il numero delle componenti si osservano:

- le componenti con autovalore maggiore di 1.
- le componenti necessarie per raggiungere il 75% di varianza spiegata.
- il punto di gomito nello screeplot, ossia l'autovalore della componente principale dopo la quale gli autovalori delle componenti principali iniziano ad assomigliarsi.

FIGURA 3.3: Scree plot della PCA sui dati normalizzati. In blu è evidenziato il punto corrispondente alla seconda componente principale.



Nonostante il punto di gomito nel grafico 3.3 indichi che sia necessario selezionare le prime tre componenti principali, gli altri due metodi di selezioni ci suggeriscono che ne siano sufficienti due. Dunque, seguendo queste indicazioni, si sceglie di tenere le prime due componenti principali.
Si può notare dalla matrice di correlazione tra le componenti principali e le variabili (Tab. 3.2), che le componenti scelte sono correlate alla varianza della curva della luce emessa dalle stelle, mentre sono quasi insensibili alle altre caratteristiche misurate.
Questo suggerisce che le componenti principali ottenute dall'analisi non riescono a rappresentare in modo completo il dataset di partenza. Sebbene abbiano catturato alcune delle variazioni nei dati, rimangono incapaci di acquisire tutte le complesse interazioni e strutture presenti nell'insieme di dati originale. Ovvero, non riescono a rappresentare tutte le sfumature del dataset.

Tabella 3.1: Output dell'algoritmo di selezione delle variabili.

Variabile	Score
LogPeriod	0.10405013
JmK	0.07727416
JmH	0.06868109
VI_Color	0.06858502
Raw_PercentileRange10	0.06681750
Raw_WeightedSkewness	0.06359030
A11	0.05627584
BV_Color	0.05581527
LogAmplitude	0.05401934
Raw_WeightedStdDev	0.04307266
nonQsoVar	0.03804404
logA12_A11	0.03498320
qsoVar	0.03137740

Tabella 3.2: Matrice di correlazione tra le prime due componenti principali e le variabili.

Variabili	Prima componente	Seconda componente
LogPeriod	-0.21756572	0.5849514
JmK	-0.25856892	0.5327005
JmH	-0.26882762	0.5528944
VI_Color	-0.24608931	0.5238849
Raw_PercentileRange10	0.35877991	-0.3187744
Raw_WeightedSkewness	-0.09999385	-0.1288475
A11	-0.43148685	0.3948248
BV_Color	-0.20021935	0.5431569
LogAmplitude	-0.50349138	0.3145906
Raw_WeightedStdDev	-0.47356489	0.4148999
nonQsoVar	-0.27069239	0.9617299
logA12_A11	-0.13193865	-0.1241737
qsoVar	-0.96284539	-0.2696967

3.2 Metodi di clustering tradizionali

3.2.1 Metodo di partizione: K-means

Dopo aver preparato il dataset, si è effettuata l'analisi dei cluster a partire dall'algoritmo K-means. Si è cercato di capire se potessero essere presenti nei dati 16 gruppi, visto che

il dataset originale è composto da 16 famiglie di stelle variabili, a loro volte suddivise in 23 categorie distinte.

TABELLA 3.3: Silhouette media e indice di Rand aggiustato calcolati sulle previsioni ottenute con K-means assumendo 16 gruppi nei dati.

Numero di cluster	Dati originali	Dati Normalizzati	Componenti principali
silhouette media	0.4479	0.4479	0.4513
ARI	0.0565	0.0562	0.0545

I risultati ottenuti (Tab. 3.3) mostrano che l'algoritmo K-means non sembra essere in grado di identificare in modo ottimale i cluster nei dati astronomici delle stelle variabili. Questo potrebbe essere attribuito al fatto che nel caso considerato, le assunzioni dell'algoritmo, ovvero che i cluster siano di forma sferica e abbiano dimensioni simili, non siano rispettate.

La rappresentazione della silhouette calcolata in ogni punto del dataset (Fig. 3.4) e la tabella 3.4, confrontata con la distribuzione dei gruppi di stelle variabili (Tab. 1.2), mostrano che il miglior risultato ottenuto con l'algoritmo K-means non riesce a individuare i cluster in modo ideale. Infatti, nella tabella 3.4 si nota che l'algoritmo non riesce ad identificare la classe di appartenenza delle stelle e considera la maggior parte di esse membre di un unico gruppo. A conferma di tale ipotesi, l'indice di Rand aggiustato, assume valore 0 (Tab. 3.3).

Dato che la distribuzione dei dati non cambia significativamente dopo la normalizzazione, dalla tabella 3.3 si può inoltre notare che si ottengono risultati identici con i dati normalizzati e non.

FIGURA 3.4: Silhouette ottenute con l'algoritmo K-means (applicato sulle prime due componenti principali).

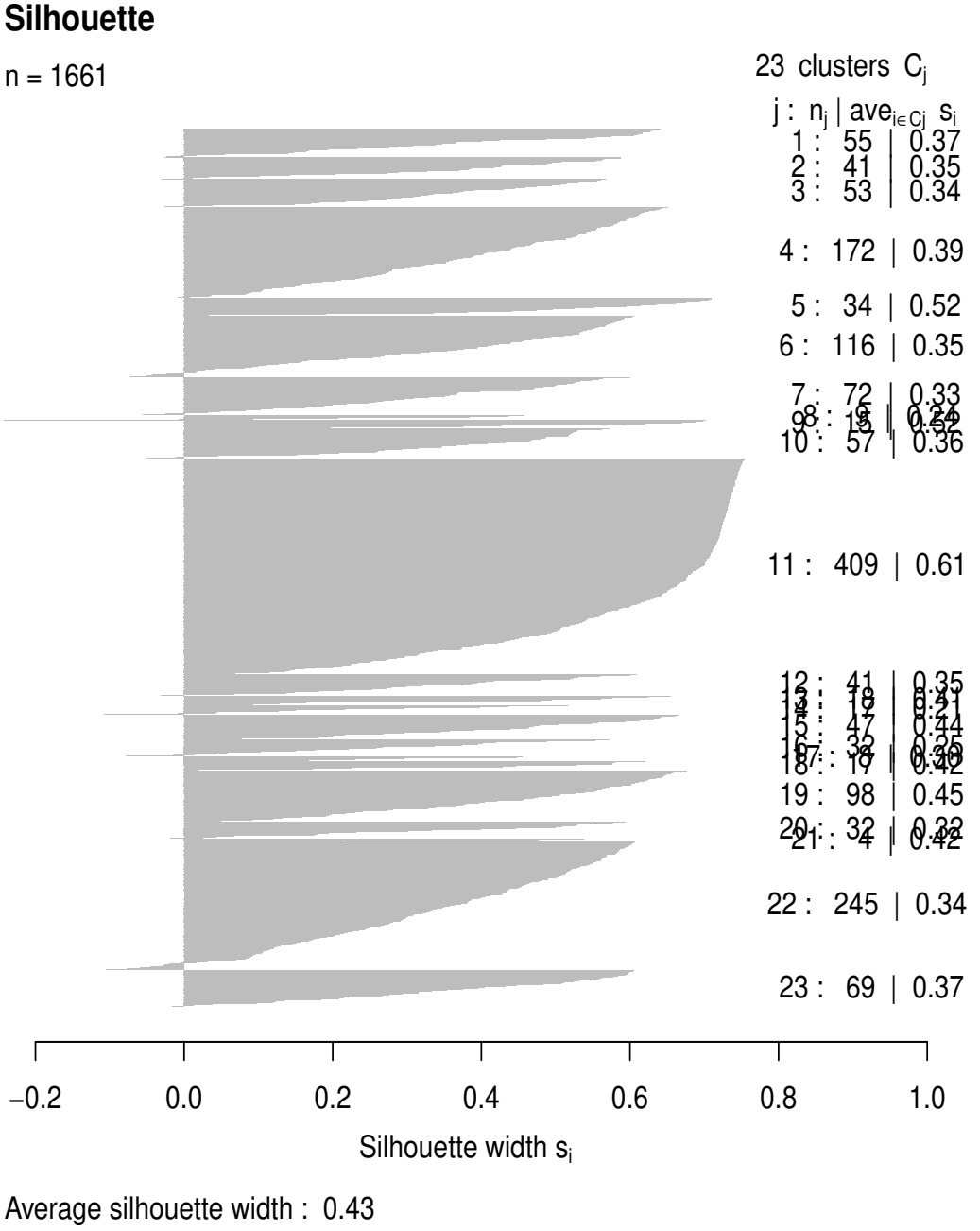


TABELLA 3.4: Previsioni dei cluster ottenuti con l'algoritmo K-means (applicato sulle prime due componenti principali) confrontate con le 16 famiglie di stelle variabili.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ACV	71	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0
ACYG	13	0	0	0	0	0	0	0	0	0	4	0	0	0	0	1
BCEP	24	0	0	2	0	0	0	0	0	1	3	0	0	0	0	0
BE+GCAS	11	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
DCEP+DCEPS+CEP(B)	6	0	4	2	16	4	0	1	2	59	95	0	0	16	0	26
CWA+CWB	0	0	0	1	0	0	0	0	0	8	4	0	0	2	0	0
DSCT+DSCTC	91	2	0	2	0	2	1	0	2	6	15	0	0	7	0	0
EA+EB+EW	154	12	0	27	1	23	7	21	12	99	149	4	1	77	0	3
ELL	19	0	0	0	0	0	0	0	0	1	5	0	0	2	0	0
GDOR	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LPV	0	2	22	22	54	28	0	26	14	7	10	0	12	15	29	44
RRAB+RRC	3	0	0	17	0	11	0	8	6	17	8	0	0	22	0	0
RS+BY	19	0	1	0	0	0	0	0	0	0	10	0	0	0	0	5
RV	0	0	0	0	0	0	0	0	0	1	2	0	0	2	0	0
SPB	80	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
SXARI	6	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

3.2.2 Metodo gerarchico: h-clust

Dopo aver definito la distanza euclidea per determinare come i punti nel dataset vengono misurati in termini di similarità, l'analisi con l'algoritmo di clustering agglomerativo ha portato a esiti preferibili, in termini di silhouette media, all'algoritmo K-means.
Al variare del legame, che determina come vengono calcolate le distanze tra i cluster durante il processo di unione, i risultati ottenuti non differiscono di molto, ad esclusione del legame di Ward che fa ottenere risultati peggiori. Probabilmente poichè con il legame di Ward si tende a minimizzare la varianza all'interno dei cluster così da creare cluster di dimensioni simili e compatti, ovvero non riesce a riflettere bene la struttura complessa dei dati. Ad ogni modo, i risultati migliori si ottengono con il legame completo (Tab. 3.5).

TABELLA 3.5: Silhouette media e indice di Rand aggiustato calcolati sulle previsioni ottenute con h-clust con collegamento completo assumendo 16 gruppi nei dati.

Numero di cluster	Dati originali	Dati Normalizzati	Componenti principali
silhouette media	0.5826	0.5826	0.6046
ARI	0.0145	0.0145	0.0104

Come per il caso precedente, dalla tabella 3.5 si può notare che si ottengono risultati identici con i dati normalizzati e non.

Il fatto che si ottengano buoni risultati con il collegamento completo, che è il metodo meno sensibile agli outliers, fa pensare che i cluster presenti nei dati siano dalle dimensioni irregolari e di dimensioni differenti tra loro. La rappresentazione della silhouette calcolata in ogni punto del dataset (Fig. 3.5) e la tabella 3.6 confrontata con la distribuzione delle famiglie di stelle variabili (Tab. 1.2) mostrano che, in realtà, il miglior risultato ottenuto con l'algoritmo h-clust non riesce a individuare i cluster in modo ideale.
Difatti, il terzo cluster individuato ha numerosità pari a 1327, ma non sono presenti gruppi così numerosi nei dati. Questo evidenzia che molti gruppi di stelle variabili risultano di numero inferiore rispetto alla realtà.
Ciò significa che la silhouette media, in questo caso, è un'indicatore distorto, probabilmente a causa dell'asimetria nelle densità dei cluster e della poca omogeneità all'interno

dei gruppi. Invece, l'indice di Rand aggiustato, he assume un valore pari a 0.104, è coerente con il fatto che h-clust non riesca ad identificare i gruppi latenti nei dati.

FIGURA 3.5: Silhouette ottenute con l'algoritmo h-clust (applicato ai dati normalizzati con collegamento medio assunte 16 famiglie di stelle variabili).

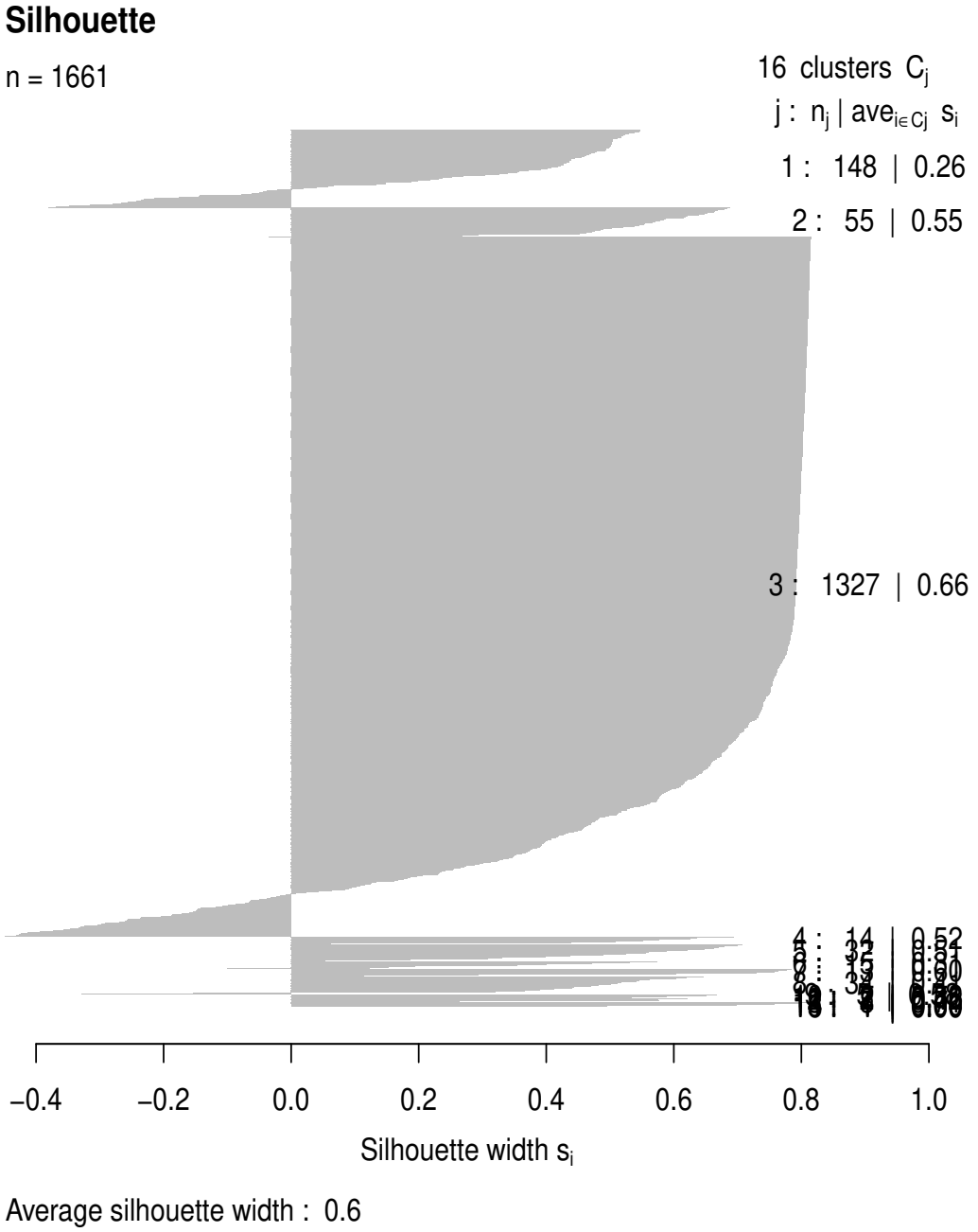


Tabella 3.6: Previsioni dei cluster ottenuti con l'algoritmo h-clust (applicato alle componenti principali con collegamento completo) confrontate con le 16 famiglie di stelle variabili.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ACV	0	0	77	0	0	0	0	0	0	0	0	0	0	0	0	0
ACYG	0	0	18	0	0	0	0	0	0	0	0	0	0	0	0	0
BCEP	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0
BE+GCAS	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0
DCEP+DCEPS+CEP(B)	6	1	216	0	2	0	0	6	0	0	0	0	0	0	0	0
CWA+CWB	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0
DSCT+DSCTC	4	0	119	0	2	0	2	0	0	1	0	0	0	0	0	0
EA+EB+EW	37	22	496	1	11	0	10	0	0	1	0	2	7	1	1	1
ELL	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0
GDOR	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0
LPV	80	24	105	13	11	15	1	27	5	0	3	0	1	0	0	0
RRAB+RRC	21	8	57	0	6	0	0	0	0	0	0	0	0	0	0	0
RS+BY	0	0	34	0	0	0	0	1	0	0	0	0	0	0	0	0
RV	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0
SPB	0	0	81	0	0	0	0	0	0	0	0	0	0	0	0	0
SXARI	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0

3.3 Metodo di Deep Learning

3.3.1 Clustering con reti neurali multistrato

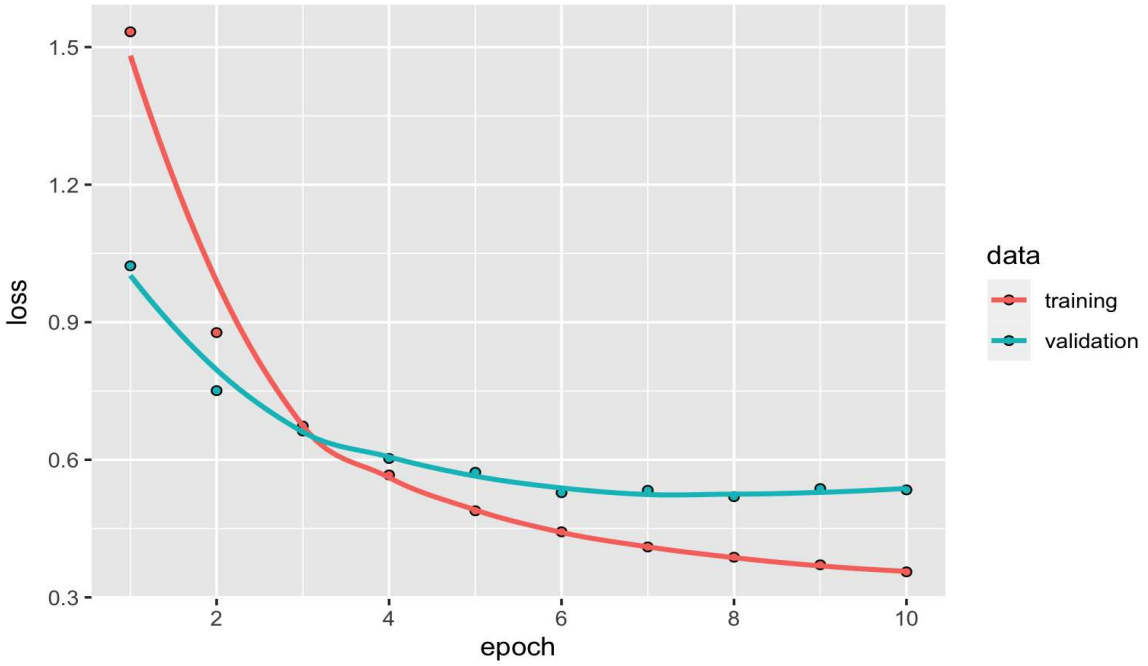
Come descritto nel capitolo 2, le reti neurali hanno la capacità di apprendere rappresentazioni latenti dei dati, la quale può essere sfruttata per eseguire il clustering dei dati. Per ciascun insieme di dati (normalizzati e componenti principali) si è adoperata la stessa struttura neurale, ovvero una rete a 6 strati. Per ogni neurone si è adottata la funzione di attivazione sigmoide, ad eccezione dei neuroni che costituiscono lo strato di output, i quali sono stati dotati di una funzione di attivazione softmax. Con lo scopo di evitare l'overfitting, le reti neurali sono state allenate per 10 epoche in una sezione pari al 70% del dataset originale.

Dato che la percentuale dei dati destinata all'allenamento non era sufficiente per creare una rete neurale capace di ottenere una rappresentazione latente dei dati, si è utilizzata una tecnica di data augmentation. Ovvero si sono copiati i dati riservati all'allenamento della rete e sono stati duplicati iterativamente per 9 volte, creando una sequenza di duplicazioni. La strategia scelta può incentivare l'overfitting, ma questo è stato mitigato dal fatto che la rete è stata addestrata in poche epoche.
Ne si ha la conferma osservando il grafico 3.6 che mostra l'andamento della funzione di perdita calcolata dalla rete neurale sui dati di addestramento e di testing. Infatti, a ridosso delle ultime epoche, la funzione di perdita mostra un'andamento costante quando calcolata sui dati di testing.

Tabella 3.7: Silhouette media e indice di Rand aggiustato calcolati sulle previsioni ottenute con le reti neurali multistrato.

Numero di cluster	Dati Normalizzati	Componenti principali
silhouette media	0.7656	0.5157
ARI	0.8221	0.4312

FIGURA 3.6: Funzione di perdita calcolata dalla rete neurale sui dati di addestramento e di testing (applicata ai dati normalizzati con 16 gruppi di stelle variabili).



Osservando la tabella 3.7 si nota che il clustering effettuato sulle componenti principali non identifica in modo ottimale la presenza dei gruppi, a differenza degli altri due algoritmi. Probabilmente perchè anche se le componenti principali non rappresentano bene il dataset, K-means e h-clust non sono abbastanza sensibili da ricavare più informazioni dai dati originali, così raggiungono la stessa qualità di clustering con le diverse forme dei dati.

In termini di silhouette media e dell'indice di Rand aggiustato, con il clustering effettutato sui dati normalizzati si ottengono i risultati migliori (Tab. 3.7). Osservando il grafico delle silhouette (Fig. 3.7) si può notare che, rispetto agli altri algoritmi di clustering, le reti neurali riescono ad identificare nel modo migliore i gruppi presenti nei dati. Infatti, la maggior parte delle osservazioni è stata assegnata al cluster corretto, questo si può vedere grazie al fatto che i valori della silhouette per la maggior parte delle osservazioni sono vicini a 1. Esaminando attentamente il grafico e la tabella 3.8 si nota però che vengono identificati 11 gruppi invece che 16. Questo potrebbe essere dovuto dal fatto che le osservazioni su cui si testa la rete sono soltanto il 30% di tutto il dataset e, a causa del forte sbilanciamento sulla numerosità dei gruppi, alcuni di questi potrebbero essere scarsamente rappresentati. Per esempio notiamo che nel sotto insieme dei dati riservati alla validazione della rete neurale, sono presenti 15 cluster.

Ripetendo l'analisi con un'indicizzazione differente per il dataset di allenamento e per quello di validazione si va incontro allo stesso problema. Si può dunque concludere che le reti neurali, a causa della numerosità dei gruppi fortemente sbilanciate (Tab. 1.2), fanno fatica ad individuare tutti i cluster presenti nei dati.

Nonostante ciò, con le reti neurali multistrato si identificano in modo più efficace i gruppi latenti nei dati quando si utilizzano i dati normalizzati. Questo, probabilmente, è dovuto al fatto che le reti non fanno assunzioni specifiche sulla forma dei cluster e quindi possono identificare gruppi dalla forma complessa.

FIGURA 3.7: Silhouette ottenute dalla rete neurale
(applicata ai dati normalizzati con 16 gruppi di stelle variabili).

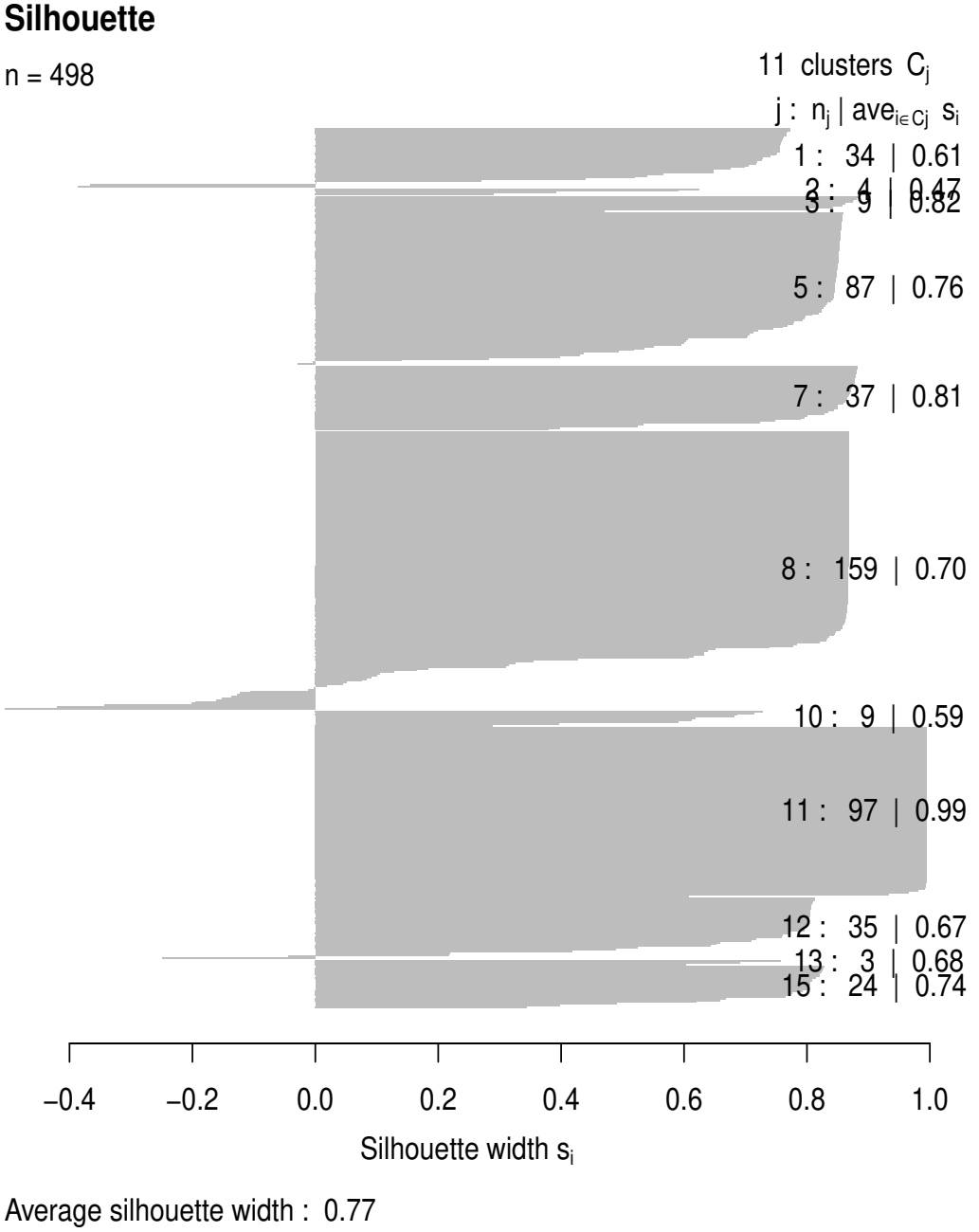


Tabella 3.8: Matrce di confusione ottenuta con la rete neurale (applicata ai dati normalizzati) a confronto con i dati di testing assunte 16 famiglie di stelle variabili.

	Osservazioni														
Previsioni	1	2	3	4	5	6	7	8	9	10	11	12	13	15	16
1	18	3	0	1	0	0	0	2	1	0	0	0	0	9	0
2	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0
3	0	0	7	0	0	0	1	0	1	0	0	0	0	0	0
5	0	1	0	0	74	2	0	4	1	0	0	0	5	0	0
7	0	0	3	0	0	0	27	6	1	0	0	0	0	0	0
8	1	0	0	0	2	2	2	142	4	0	0	4	1	1	0
10	1	0	0	0	0	0	0	2	1	3	0	2	0	0	0
11	0	0	0	0	0	0	0	0	0	0	97	0	0	0	0
12	0	1	0	0	2	0	0	10	1	0	0	21	0	0	0
13	0	0	0	0	0	0	0	0	1	0	0	0	2	0	0
15	3	0	0	1	0	0	0	3	0	0	0	0	0	15	2

Capitolo 4

Conclusioni

Con la crescente disponibilità di dati riguardanti il campo astronomico, avere metodologie analitiche in grado di estrarre informazioni da dateset complessi è l'esigenza del momento. In questa tesi si sono confrontati due algoritmi di clustering tradizionali (K-means e h-clust) con uno basato sul Deep Learning (reti neurali multistrato) al fine di determinare l'efficacia di quest'ultimo come metodologia innovativa per il clustering di stelle variabili. Come parametri per valutare la qualità della clusterizzazione sono state considerate la separazione dei cluster e la capacità degli algoritmi di rilevare correttamente le stelle presenti nel dataset.

Tabella 4.1: Migliori risultati per ciascun algoritmo ipotizzando 16 cluster. Con K-means e h-clust ottenuti sulle componenti principali, con le reti neurali ottenuto con i dati normalizzati.

Algoritmo	K-means	h-clust	Reti neurali
Silhouette media	0.4513	0.6046	0.7656
ARI	0.0545	0.0104	0.8221

- Con K-means si ottiene un raggruppamento delle stelle poco rappresentativo a causa delle assunzioni dell'algoritmo: i cluster latenti nei dati non sono di forma sferica e con dimensioni simili tra loro. Questo lo si può osservare anche nella tabella 1.2, la quale evidenzia la natura sbilanciata delle classi di stelle variabili.
- Durante l'analisi con l'algoritmo h-clust si sono provate diversi metriche di legame, a scopo di interpretare la forma dei cluster e con il legame completo si sono ottenuti i risultati migliori. Questo suggerisce che i cluster siano ben distanziati tra loro, che siano presenti degli outliers e che la forma dei cluster sia dunque irregolare. Andando a vedere la silhouette (Fig. 3.5) e l'indice di Rand aggiustato,

ci si rende conto che l'algoritmo non distingue correttamente i gruppi di stelle. Questo probabilmente per via del fatto che i gruppi latenti nei dati hanno densità asimmetriche e sono poco distanziati tra loro.

- L'analisi per mezzo delle reti neurali ha potrato ad ottenere la miglior rappresentazione dei gruppi presenti nei dati come si può osservare dal valore della silhouette media e da quello dell'indice di Rand aggiustato (Tab. 4.1).
  I valori nel grafico della silhouette (Fig. 3.7) mostrano chiaramente che le reti neurali, sebbene non siano state in grado di identificare tutti i gruppi nel dataset in modo ottimale, sono riuscite a distinguere efficacemente la separazione tra i cluster riconosciuti. Questo indica una buona coesione all'interno dei gruppi, dovuta alla capacità dell'algoritmo di rappresentare forme più complesse dei cluster a seconda dell'architettura e del training.
  Il fatto che le reti neurali non riescano ad individuare tutti e 16 i gruppi, ma soltanto 11, è dovuto al fatto che alcuni gruppi sono scarsamente rappresentati nel test set. Con una maggiore quantità di dati probabilmente si otterrebbe un'identificazione dei gruppi più completa.

In aggiunta, si è potuto osservare che l'algoritmo K-means e h-clust identificano meglio i gruppi presenti nei dati quando applicati sulle componenti principali, ma si è visto che queste non riescono a rappresentare integralmente il dataset. I migliori risultati di h-clust e K-means che vengono ottenuti con le componenti principali sono dovuti al fatto che questi due algoritmi preferiscono dataset dalla dimensionalità ridotta ed al fatto che le componenti principali possono fornire caratteristiche dei dati più informative, aiutandoli a individuare i cluster nei dati.
D'altro canto, normalizzare i dati e mantenere la relazione tra le variabili è fondamentale quando si utilizzano reti neurali, poiché aiuta a migliorare la stabilità e l'efficienza dell'addestramento.

Si può dunque concludere che nel campo astronomico, con riferimento specifico al caso delle stelle variabili, le reti neurali multistrato sono una metodologia di clustering più efficace rispetto a quelle tradizionali. Una problematica da sottolineare però è il carico computazionale che richiedono le reti neurali per il loro addestramento, di gran lunga superiore rispetto agli algoritmi di clustering tradizionali.
Il passo successivo potrebbe essere quello di ottimizzare la struttura della rete per dataset

come i cataloghi stellari e l'apprendimento su dataset sbilanciati, dato che le reti neurali richiedono grandi quantità di dati per l'addestramento efficace.

Bibliografia

Aitchison, J. (1985). A general class of distributions on the simplex. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 47(1):136–146.

Berry, M. W., Mohamed, A., and Yap, B. W. (2019). *Supervised and Unsupervised Learning for Data Science*. Springer.

Dubath, P., Rimoldini, L., Süveges, M., Blomme, J., López, M., Sarro, L., De Ridder, J., Cuypers, J., Guy, L., Lecoeur, I., et al. (2011). Random forest automated supervised classification of hipparcos periodic variable stars. *Monthly Notices of the Royal Astronomical Society*, 414(3):2602–2617.

Dudek, A. (2020). Silhouette index as clustering evaluation tool. In *Classification and Data Analysis: Theory and Applications 28*, pages 19–33. Springer.

Eker, Z., Ak, N. F., Bilir, S., Doğru, D., Tüysüz, M., Soydugan, E., Bakış, H., Uğraş, B., Soydugan, F., Erdem, A., et al. (2008). A catalogue of chromospherically active binary stars. *Monthly Notices of the Royal Astronomical Society*, 389(4):1722–1726.

Genuer, R., Poggi, J.-M., and Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225–2236.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2:193–218.

Kurita, T. (2019). Principal component analysis (pca). *Computer Vision: A Reference Guide*, pages 1–4.

Madhulatha, T. S. (2012). An overview on clustering methods. *arXiv preprint arXiv:1205.1117*.

Montufar, G. F., Pascanu, R., Cho, K., and Bengio, Y. (2014). On the number of linear regions of deep neural networks. *Advances In Neural Information Processing systems*, 27.

Parmar, A., Katariya, R., and Patel, V. (2019). A review on random forest: An ensemble classifier. In *International conference on intelligent data communication technologies and internet of things (ICICI) 2018*, pages 758–763. Springer.

Percy, J. R. (2007). *Understanding variable stars*. Cambridge University Press.

Perryman, M. A., Lindegren, L., Kovalevsky, J., Hoeg, E., Bastian, U., Bernacca, P., Crézé, M., Donati, F., Grenon, M., Grewing, M., et al. (1997). The hipparcos catalogue. *Astronomy and Astrophysics, Vol. 323, p. L49-L52*, 323:L49–L52.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850.

Song, C., Huang, Y., Liu, F., Wang, Z., and Wang, L. (2014). Deep auto-encoder based clustering. *Intelligent Data Analysis*, 18(6S):S65–S76.

Van Leeuwen, F. (1997). The hipparcos mission. *Space Science Reviews*, 81(3-4):201–409.

Watson, C., Henden, A., and Price, A. (2016). Vizier online data catalog: Aavso international variable star index vsx (watson+, 2006-2014). *VizieR online data catalog*, pages B–vsx.