Stars Type Classification

Andrea Sorgente, 830483 Agostino Tassan Mazzocco, 833933

07 Luglio 2021

Abstract

Spinti dall'interesse per l'argomento e dalla curiosità che esso suscita, questo studio si pone l'obiettivo di elaborare un modello capace di identificare la tipologia di stella data una serie di sue caratteristiche. Il dataset e le relative informazioni sono state ottenute su kaggle. Per il raggiungimento dell'obiettivo è stato fatto ricorso a svariate tecniche di Machine Learning, sia supervisionate, per classificare appunto ciascuna osservazione, sia non supervisionate, con l'ambizioso scopo di trovare relazioni 'nascoste' trai dati. I risultati sono stati particolarmente incoraggianti, consentendoci di fare importanti considerazioni sul tipo di dataset preso in analisi.

Introduzione

Negli ultimi decenni sono stati fatti passi in avanti nel campo della classificazione stellare, che in astronomia è la classificazione delle stelle sulla base del loro spettro, per cui il presente elaborato si pone come obiettivo quello di considerare modelli di Machine Learning in tale ambito ampiamente studiato. Quindi siamo stati motivati alla comprensione del problema e allo svolgimento del lavoro che vedeva l'obiettivo finale di elaborare il modello più semplice che potesse prevedere commettendo il minor numero possibile di errori di classificazione su nuovi dati. Ciò è stato possibile dopo aver analizzato i dati da un punto di vista esplorativo e fatto considerazioni statistiche basate sull'analisi della correlazione tra variabili, sul tipo di variabili e riguardanti il feature scaling, oltre all'analisi delle distribuzioni di ciascuna variabile condizionata alle classi. I risultati hanno evidenziato una bassa complessità dei dati, che infatti non necessitano di modelli altamente complessi e flessibili per essere classificati correttamente. Infatti il modello finale di supervised learning che fornisce accuracy pari a 1, ovvero non vengono commessi errori di previsioni nel training set con leave-one-out cross validation e nel test set, è un albero di classificazione che sfrutta soltanto due delle sei variabili esplicative contenute nel dataset. Riteniamo dunque che le difficoltà principali legate alla classificazione stellare non siano tanto di tipo statistico/informatico nello sviluppo di algoritmi per la risoluzione del problema, quanto di tipo strumentale e scientifico nella rilevazione di dati e nell'individuazione delle variabili rilevanti basata su concetti teorici.

Materiali

Il dataset contiene le caratteristiche fisiche di 240 stelle, contenute in 7 features:

- Temperatura

La temperatura è rilevata in Kelvin, applicando la legge di Wien che consente di rilevare la temperatura di superficie della stella attraverso la lunghezza d'onda della radiazione emessa.

- Luminosità relativa

Variabile numerica che misura la luminosità relativa del corpo celeste, cioè il rapporto tra la luminosità della stella e la luminosità del sole, entrambe espresse in Watt e rilevate attraverso la legge di Stefan-Boltzmann.

- Raggio Relativo

Variabile numerica che misura il raggio relativo del corpo celeste, cioè il rapporto tra il raggio della stella e il raggio del sole, entrambi espressi in metri.

- Magnitudine Assoluta

Variabile numerica che misura la luminosità intrinseca dell'oggetto stellare.

- Colore

Variabile categoriale che ammette 7 livelli: "Red", "Orange", "Yellow", "Yellow-white", "White", "Blue-white", "Blue". Rappresenta il colore assoluto che manifesta ciascuna delle stelle.

- Classe Spettrale

Variabile categoriale ordinata che ammette 7 livelli: "M", "K", "G", "F", "A", "B", "O". Essa rappresenta le categorie di classe spettrale a cui appartiene ciascuna stella secondo la classificazione di Harvard. Le stelle di tipo O sono le più calde, le altre lettere sono assegnate a stelle via via meno calde, fino a quelle più fredde di classe M. È uso descrivere le stelle di classe O come "blu", quelle di classe B come "azzurre", quelle di classe A come "bianche", quelle di classe F come "bianco-gialle", quelle di classe G come "gialle", quelle di classe K come "arancioni" e quelle di classe M come "rosse".

- Tipo

Variabile categoriale che ammette 6 livelli: "0", "1", "2", "3", "4", "5". Ogni livello rappresenta il tipo di stella, rispettivamente Nana Bruna, Nana Rossa, Nana Bianca, Sequenza Principale, Supergigante, Ipergigante.

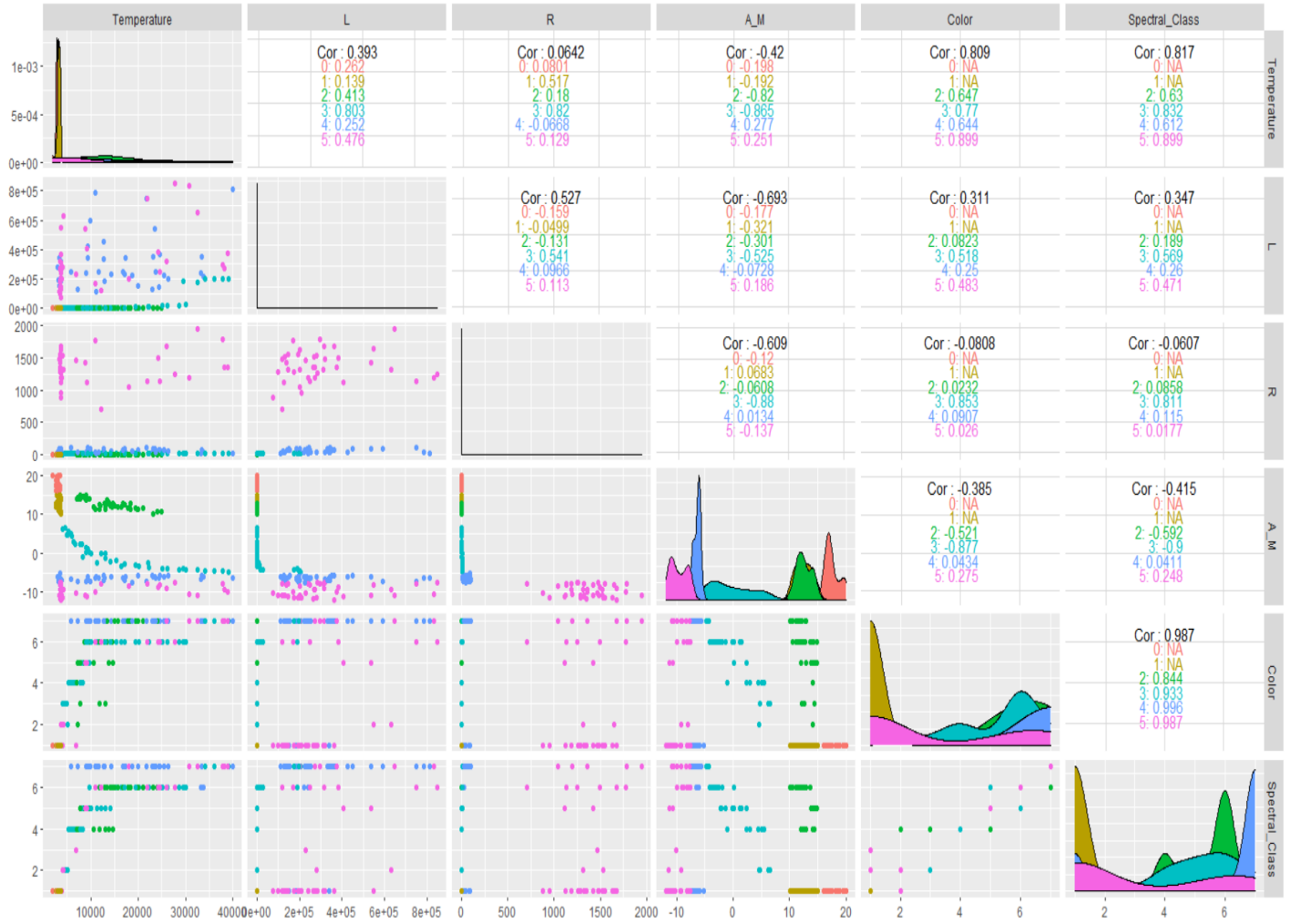
All'interno del dataset non sono presenti valori mancanti.

Metodi e Risultati

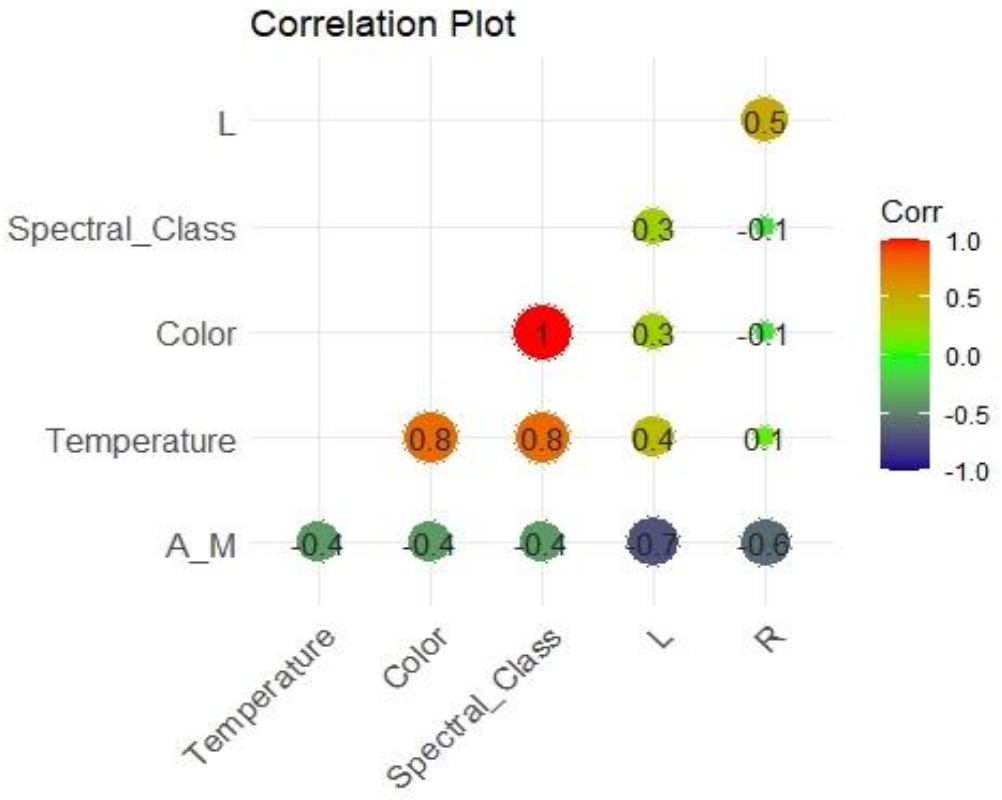
Le metodologie applicate possono essere divise in tre fasi: pre-processing, unsupervised learning e supervised learning.

Pre-processing

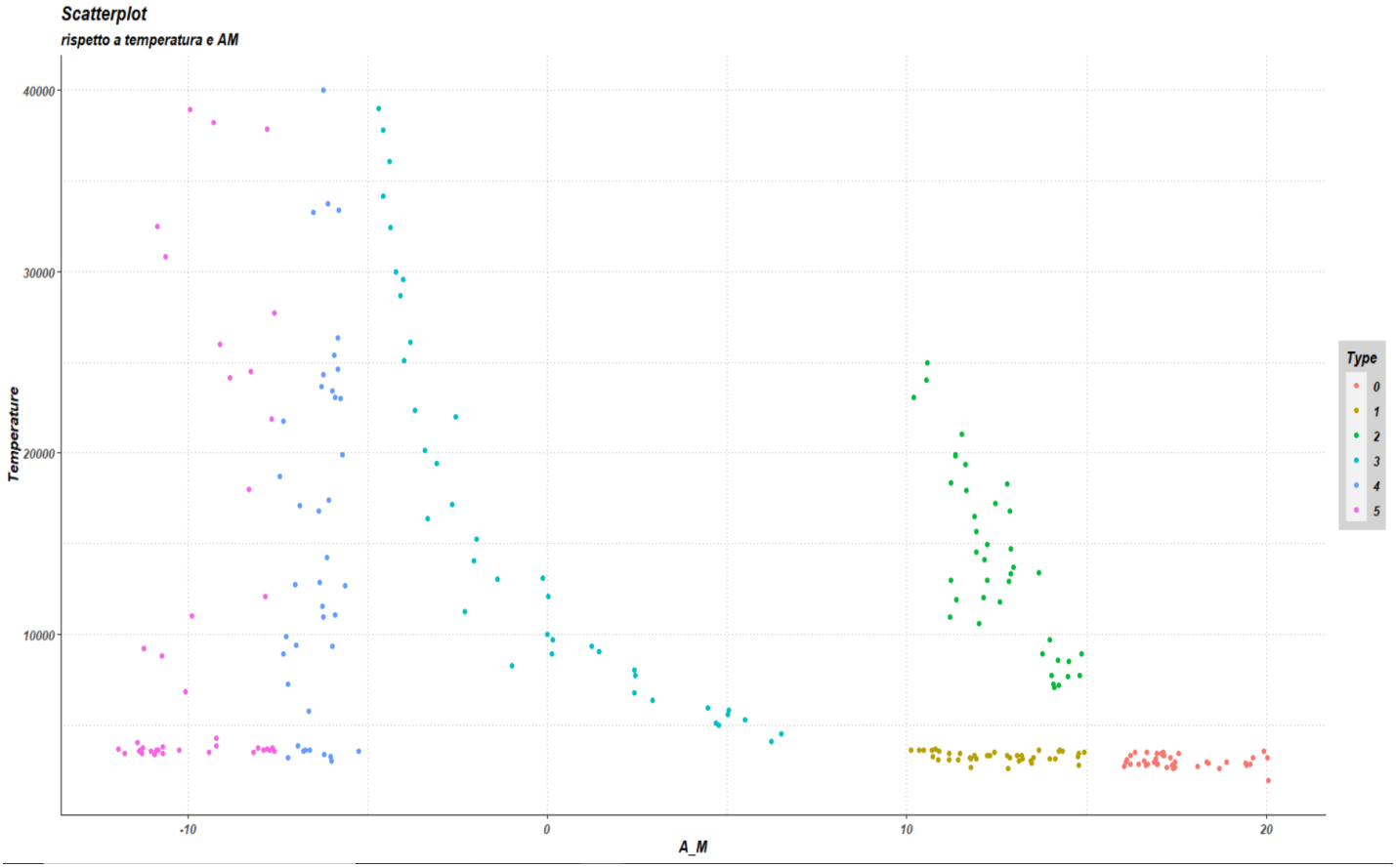
Inizialmente il dataset è stato modificato nelle due variabili categoriali "Colore" e "Classe Spettrale". I dati relativi alla variabile "Colore" si presentavano divisi in 17 livelli, la maggior parte delle quali si differenziavano per degli errori di battitura nella stringa di testo che rappresentava la categoria, pertanto è stato necessario accorpare le stesse categorie scritte in maniera diversa. Successivamente i livelli della variabile 'Colore' sono stati trasformati in valori numerici ordinati da 1 a 7, ovvero dal rosso al blu nel rispetto della classificazione di Harvard (come specificato in Materiali-Classe Spettrale). Da tale classificazione emerge chiaramente che i colori delle stelle dipendono dalla temperatura effettiva del corpo celeste, che è a sua volta strettamente correlata con la lunghezza d'onda delle radiazioni emesse. Per quanto riguarda la variabile "Classe Spettrale" è stata anch'essa trasformata in valori numerici ordinati da 1 a 7, come secondo Harvard da M a O. Risulta evidente la relazione che lega le classi spettrali degli astri al loro colore e alla loro temperatura. Il seguente grafico riporta delle informazioni importanti delle variabili presenti nel dataset:



Risulta perciò chiaro, sia dal punto di vista scientifico sia da quello statistico che le variabili 'Color' e 'Spectral Class' spieghino allo stesso modo, o quasi, la variabile risposta 'Type', infatti la loro correlazione è pari a 0.987. Inoltre la variabile 'Temperature' ha un legame simile a queste due variabili, con indice di correlazione pari a 0.809 con il colore della stella e a 0.817 con la sua classe spettrale come evidenziato nel seguente correlation plot.

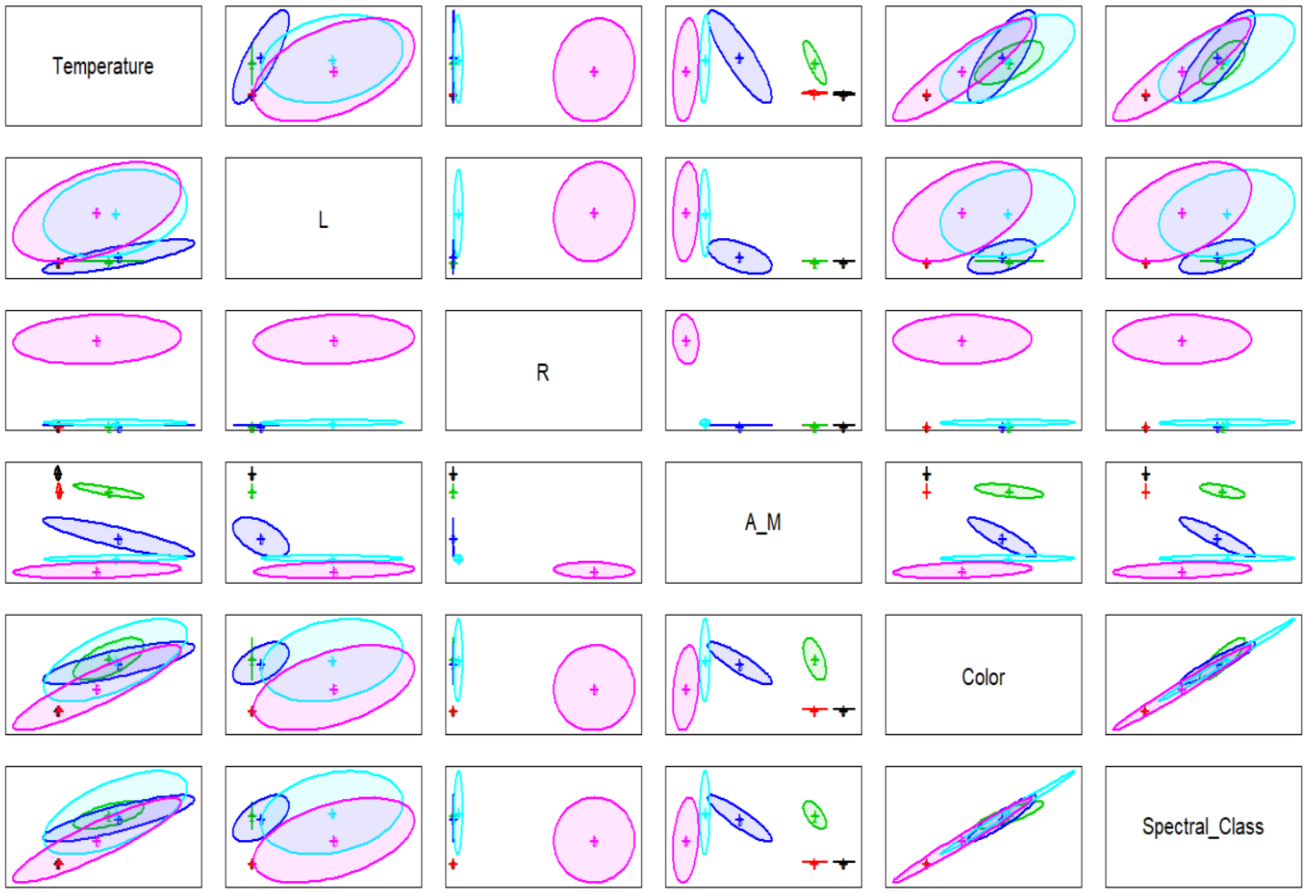


Dovendo evitare problemi di multicollinearità e tenendo conto del fatto che, almeno graficamente, la temperatura e la magnitudine assoluta sembrano essere le variabili più discriminanti per classificare le stelle, si è deciso di escludere le variabili 'Color' e 'Spectral Class' in qualsiasi modello specificato in questo report.



Inoltre notando dai successivi boxplot condizionati alle classi che le variabili contenute nel dataset hanno misure diverse, facendo anche riferimento a diverse caratteristiche di una stella, e siccome verranno considerate tecniche di clustering basate su distanza geometrica, è stato opportuno applicare una standardizzazione dell'intervallo delle variabili.

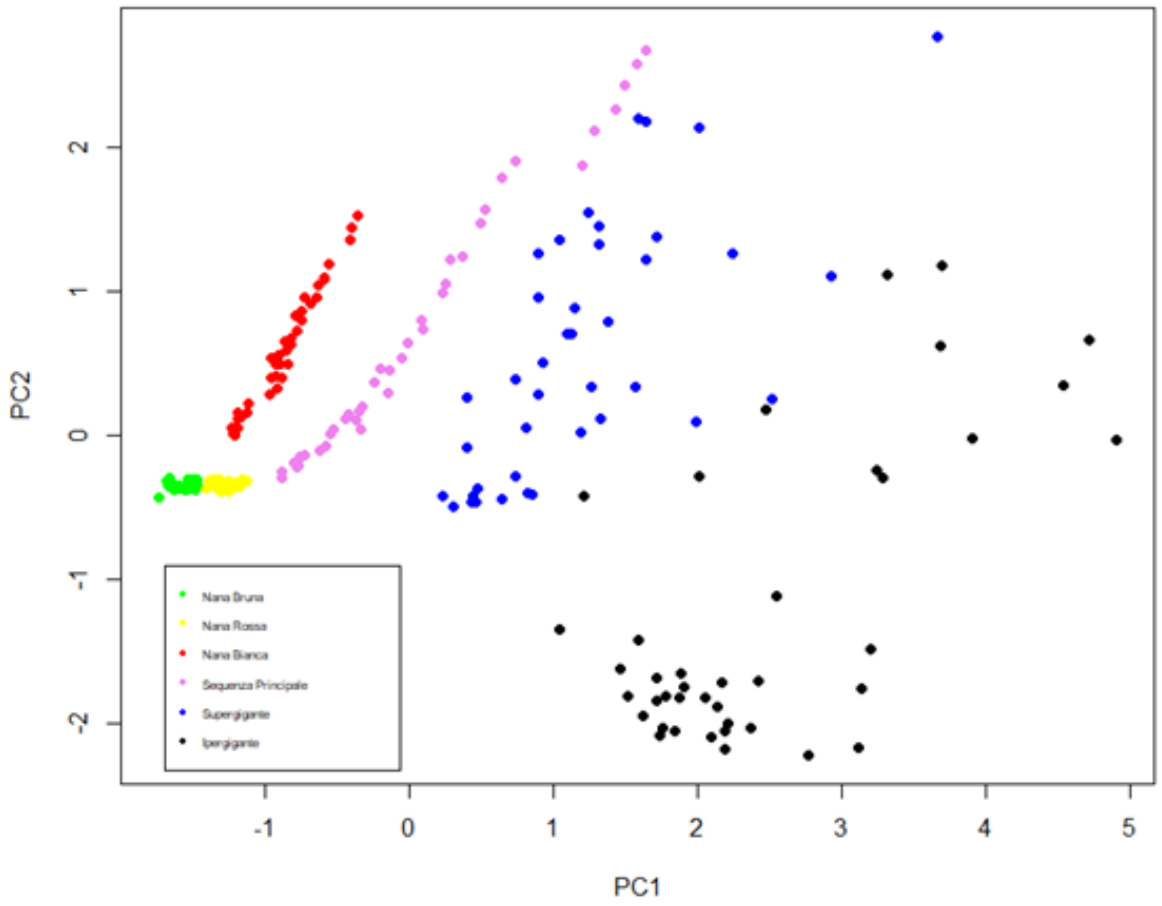
Inoltre si è voluto indagare sulla distribuzione di ciascuna variabile condizionata alla classe, ancora in maniera grafica, mediante uno strumento più efficace del boxplot. Risulta evidente come la temperatura e la magnitudine assoluta siano variabili molto discriminanti per la risposta 'Tipo', ed è altrettanto evidente come 'Color' e 'Spectral Class' abbiano una differenza minima nel potere discriminante.



Successivamente è stata eseguita la suddivisione del dataset in training set (85% delle osservazioni) e test set (restante 15%).

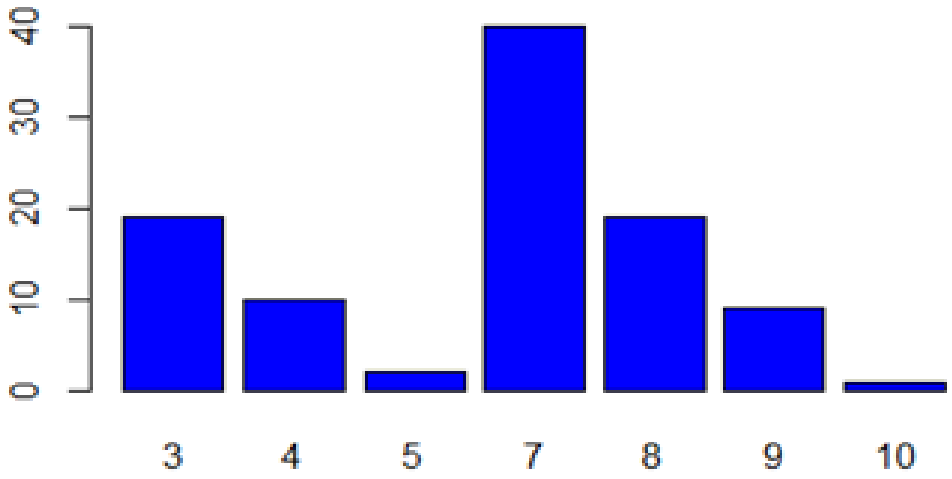
Unsupervised Learning

I dati iniziali, rappresentati nelle loro prime due componenti principali e differenziati in base alle etichette, si presentano nel modo seguente.

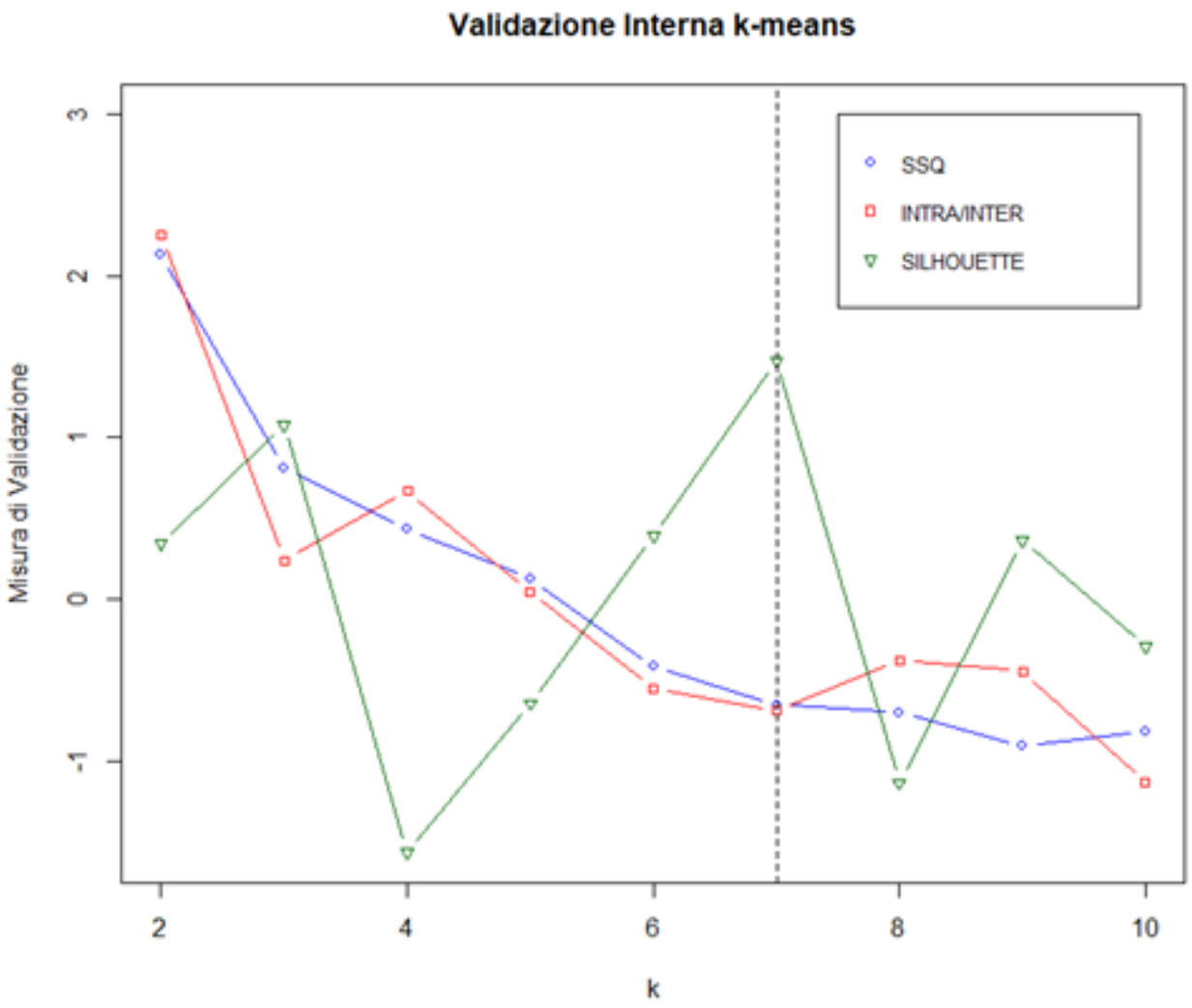


Si nota immediatamente come le classi “nana bruna” e “nana rossa” probabilmente saranno difficili da individuare, soprattutto dagli algoritmi basati sulla distanza, in quanto, nelle prime due componenti principali, condividono caratteristiche molto simili.

Il primo algoritmo applicato è il k-means. Supponendo di non conoscere a priori il numero totale delle classi, risulta necessario cercare evidenze empiriche che ci permettano di sceglierlo correttamente. La prima indicazione è stata ottenuta effettuando 100 iterazioni, per ognuna delle quali è stato utilizzato l’iperparametro $2 \leq k \leq 9$. Ad ogni iterazione è stato individuato il valore di k per cui lo score della silhouette fosse maggiore. Il risultato finale è rappresentato nel seguente grafico a barre che riporta la frequenza delle iterazioni in cui ciascun valore di k è stato contraddistinto dal maggiore valore del coefficiente di silhouette.

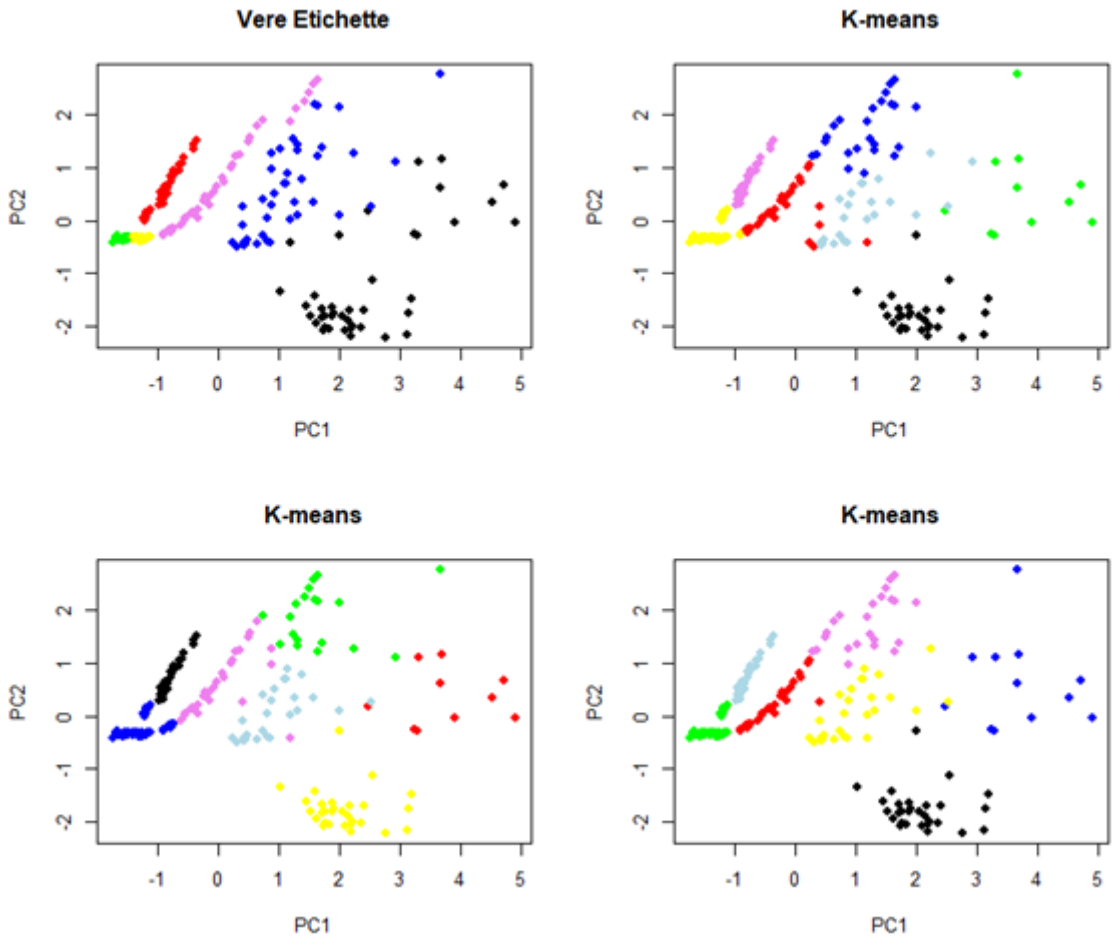


Il risultato più frequente risulta essere k=7, che non corrisponde al vero numero di classi presenti nel dataset. Per verificare che k=7 sia corretto, è stato nuovamente applicato l'algoritmo k-means per diversi valori di *k*, per ciascuno dei quali sono state calcolati i seguenti criteri di validazione interna: il silhouette coefficient, il tasso di distanza intra/inter cluster e la somma al quadrato delle distanze dai centroidi. Il risultato ottenuto è riportato nel grafico.



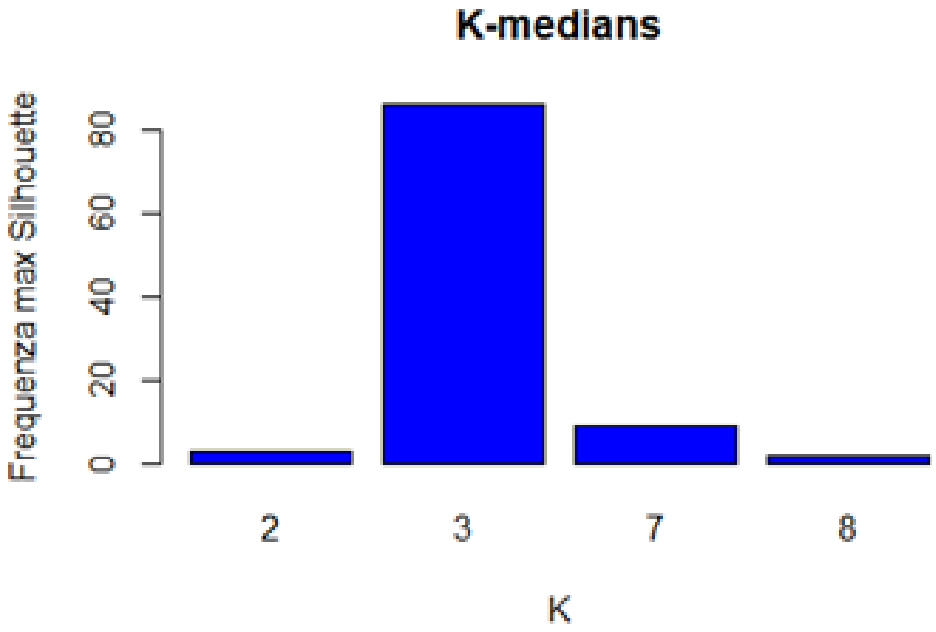
Dal grafico ottenuto è possibile notare come per quanto riguarda la silhouette, essa sia massimizzata nel punto k=7, e gli altri due coefficienti decrescono fino a k=5 e k=6 circa per poi stabilizzarsi, dunque, per la regola del gomito e per il criterio di massimizzazione del coefficiente di silhouette, sembra che k=7 sia la scelta corretta.

A questo punto è possibile applicare l'algoritmo k-means, utilizzando l'iperparametro k=7 e osservare i risultati ottenuti, rappresentati graficamente tramite le prime due componenti principali.

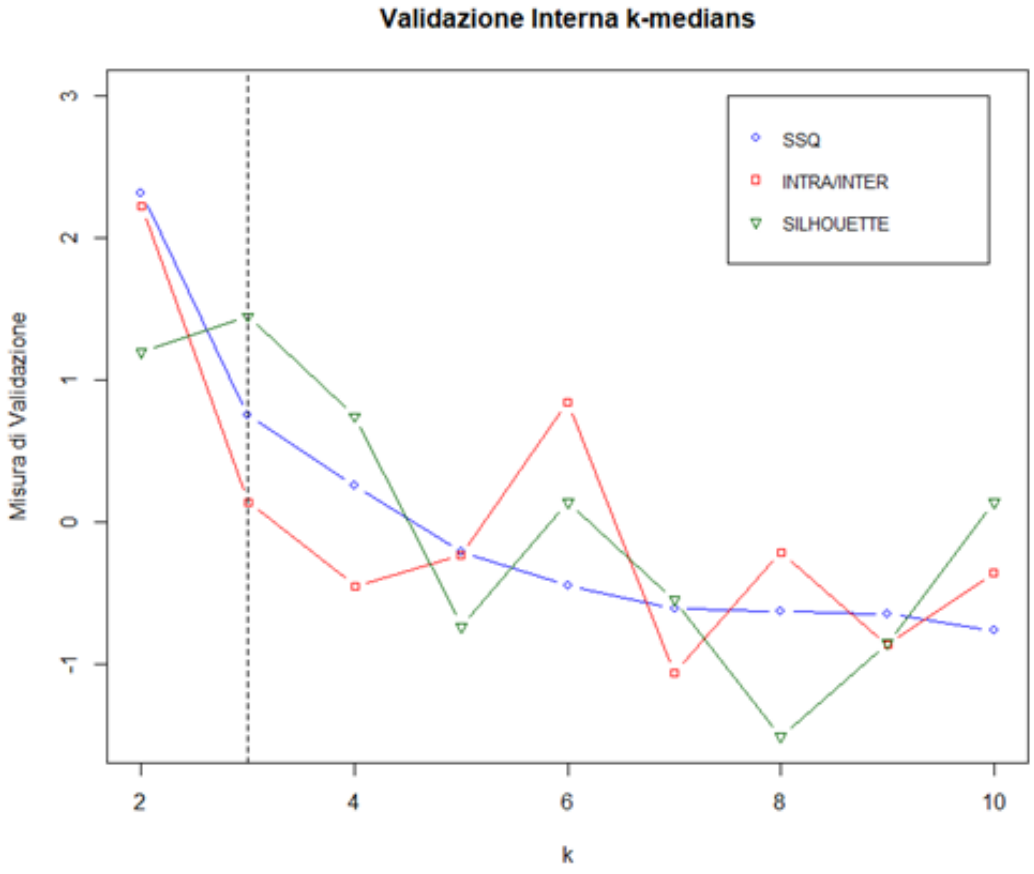


Si nota immediatamente, come era da attendersi, che le classi "nana bruna" e "nana rossa" vengono agglomerate nello stesso cluster. Risulta inoltre evidente come le stelle di sequenza principale vengano invece divise in più cluster, questo perché essi sono i corpi celesti caratterizzati da una più elevata variabilità nelle loro caratteristiche, infatti rappresentano la più lunga fase del ciclo vitale di una stella.

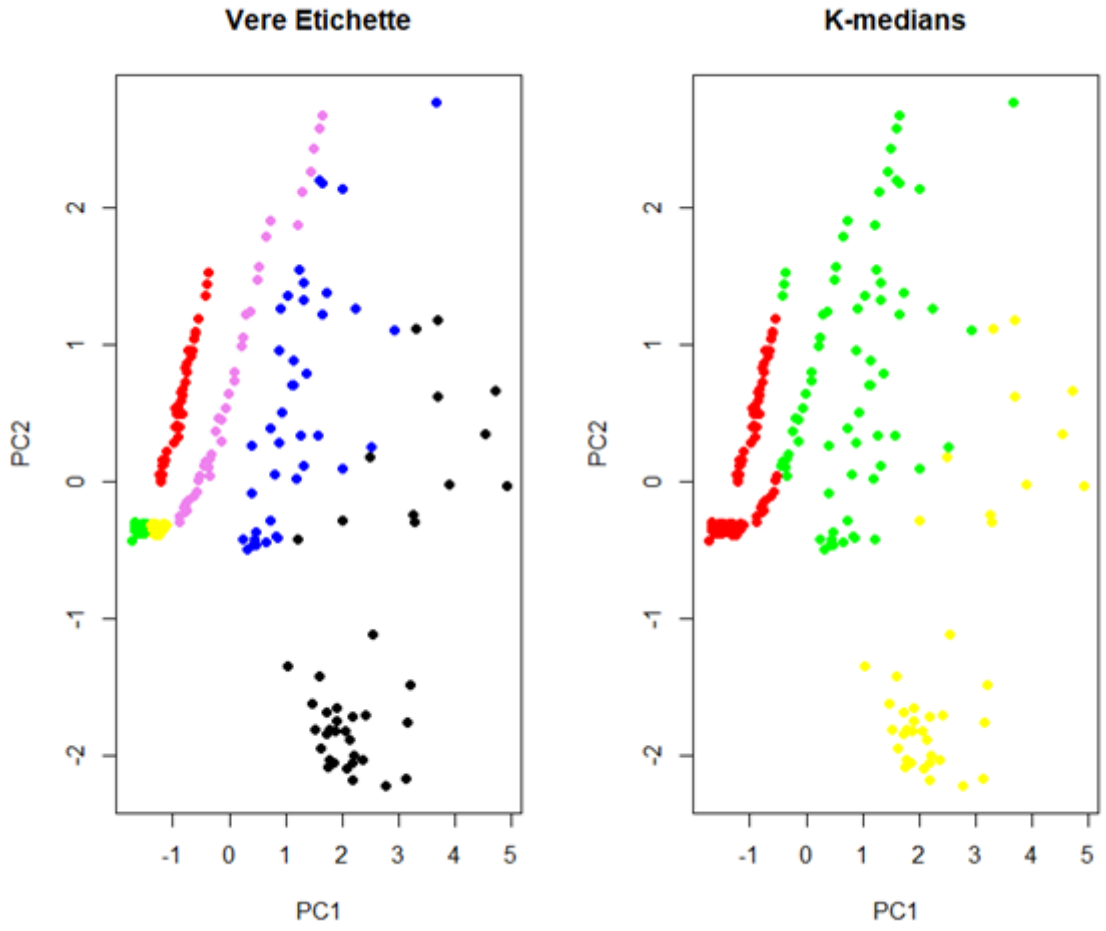
Dopo aver applicato un algoritmo di clustering basato sulle distanze euclidee (k-means), può essere utile considerare un algoritmo basato sulla distanza di Manhattan, il k-medians. Attraverso la medesima metodologia del caso precedente, il risultato più frequente, basato sul silhouette coefficient, risulta essere k=3, a differenza del k-means, da cui era stato ottenuto un valore di k=7.



Anche in questo caso tutti i criteri di validazione interna sembrano confermare il risultato ottenuto. Il silhouette coefficient è massimizzato per k=3, mentre sia la distanza intra/inter cluster che la somma al quadrato della distanza dai centroidi, sembrano presentare un punto di flessione al livello k=3. A questo punto è possibile applicare l'algoritmo k-medians, utilizzando l'iperparametro k=3.



Nel grafico è riportato il risultato dell'algoritmo k-medians con k=3, confrontato con le vere etichette.

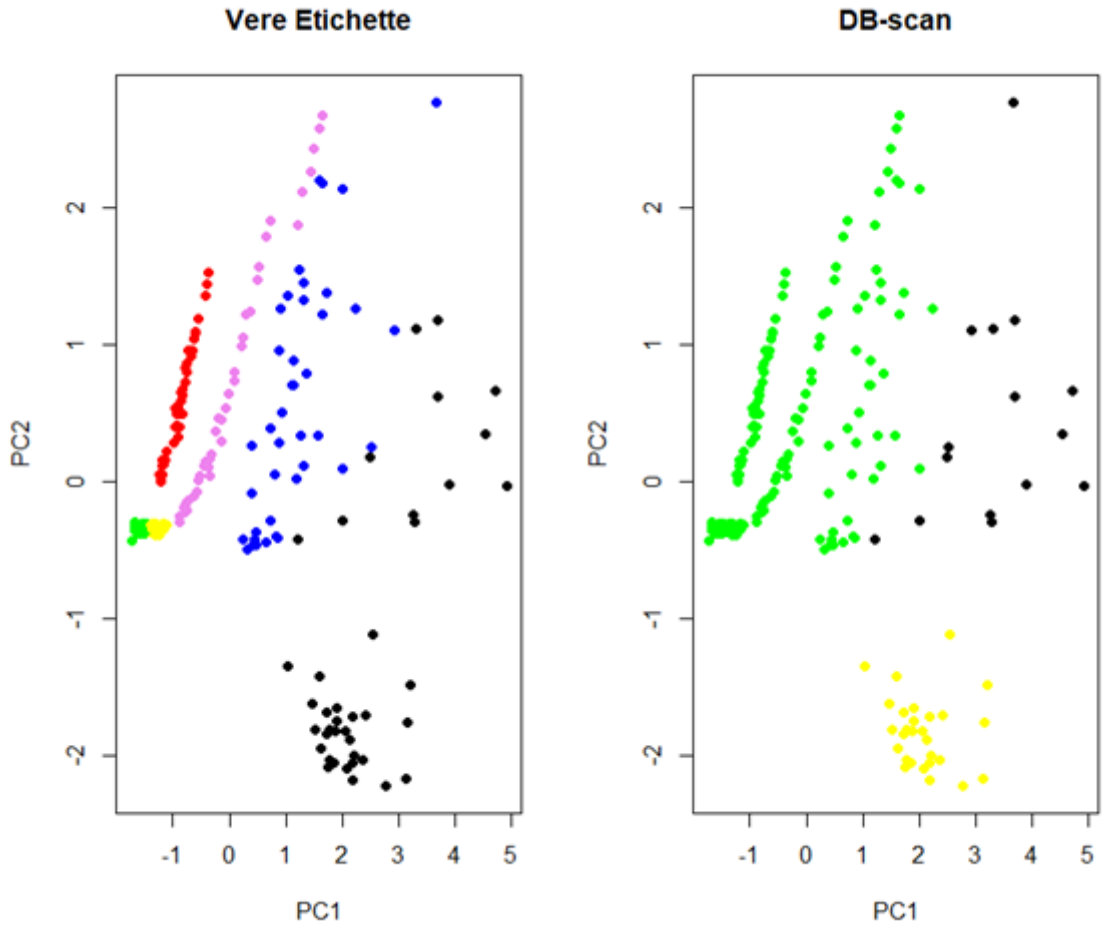


Si nota come ad eccezione di alcuni punti, le classi "nana bruna", "nana rossa" e "nana bianca" siano state agglomerate dall'algoritmo in un unico cluster, dal momento che hanno caratteristiche molto simili tra loro, soprattutto per quanto riguarda le prime due. Le stelle di sequenza principale, invece, ancora una volta sono state agglomerate alcune alle stelle nane, altre alle stelle supergiganti. Infine le stelle appartenenti alla classe "ipergiganti" sono state assegnate al medesimo gruppo, senza commettere quasi nessun errore.

Confrontando questi primi due algoritmi basati sulle distanze, si può notare come il k-means si avvicini maggiormente al vero numero di classi, senza tuttavia riuscire mai a creare dei cluster che corrispondano a quelli originali. L'algoritmo k-medians, invece, sottostima il numero di classi, ma riesce ad organizzare quasi perfettamente la classe "Ipergigante" in un unico cluster, e accorpa le altre classi nei restanti due con maggiore precisione.

A questo punto, dato che i due algoritmi basati sulla distanza non hanno fornito risultati soddisfacenti, può essere utile applicare un algoritmo basato sulla densità. In particolare è stato utilizzato l'algoritmo db-scan. Gli iperparametri dell'algoritmo, τ ed ε, sono stati scelti applicando il db-scan per una griglia di valori ($0.1 \leq \varepsilon \leq 2$ con un incremento di 0.1 e τ da 1 a 10), ed è stata

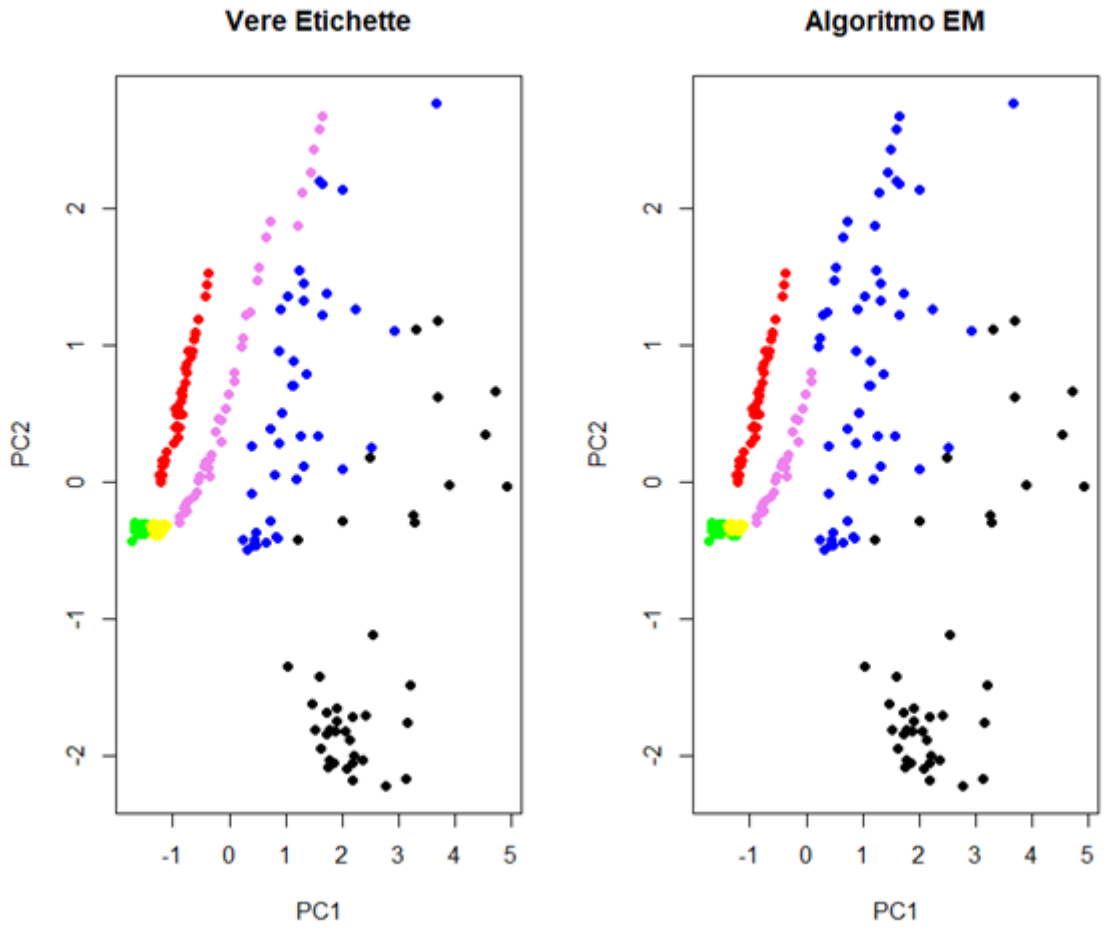
selezionata la coppia di valori che massimizzasse il silhouette coefficient, in particolare, il risultato ottenuto è stato ε=1 e τ=4.



Nel grafico è possibile osservare il risultato ottenuto. L'algoritmo ha diviso i dati in due cluster, il primo contiene la maggior parte delle osservazioni (197), il secondo contiene le restanti osservazioni eccetto i 15 noise points (rappresentati in nero). Il db-scan, se confrontato con le vere etichette, unisce in un unico cluster i dati di tutte le classi, tranne l'etichetta "ipergiganti", la quale è divisa in parte nel secondo dei due cluster e in parte nei 15 noise points. In questo caso la prestazione del db-scan è peggiore rispetto a quella dei primi due metodi basati sulla distanza.

Essendo il silhouette coefficient un criterio di validazione interna che si basa sulle distanze, potrebbe essere distorto se applicato ad algoritmi che non si basano sulla distanza, come il db-scan. Ulteriori tentativi variando gli iperparametri non hanno portato a risultati migliori.

L'ultimo metodo applicato, sfrutta un algoritmo di tipo probabilistico, in particolare l'algoritmo EM. A differenza degli algoritmi utilizzati fino a questo momento, l'algoritmo EM si è rivelato essere computazionalmente molto oneroso, nonostante il dataset sia di dimensioni ridotte. Il risultato ottenuto utilizzando come iperparametro il vero numero delle etichette, a noi noto a priori, ovvero k=6, sembra essere piuttosto soddisfacente.



Dal grafico si può vedere come l'EM sia l'unico algoritmo in grado di distinguere in due gruppi i dati le cui vere etichette sono "nana bruna" e "nana rossa". Inoltre le classi "ipergigante" e "nana bianca" sono raggruppate senza commettere errori. L'unica criticità riscontrata riguarda le stelle di classe "sequenza principale", in quanto sono state in parte accorpate alle stelle "supergiganti". In questo caso, essendo il numero di cluster pari al numero di classi delle vere etichette, è possibile compilare la matrice di confusione e valutare più dettagliatamente la prestazione di questo algoritmo. In particolare è stata rilevata un'accuratezza del 92.5%.

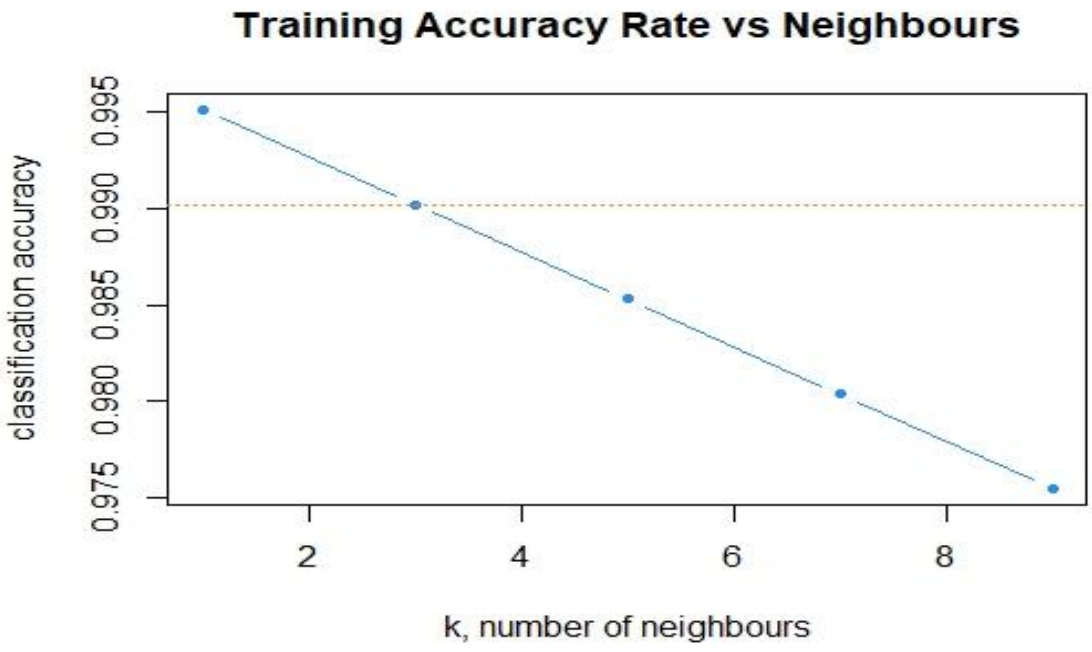
	Vere etichette						
		0	**1**	**2**	**3**	**4**	**5**
	0	38	2	0	0	0	0
Previsioni	**1**	2	38	0	0	0	0
EM	**2**	0	0	40	0	0	0
	3	0	0	0	26	0	0
	4	0	0	0	14	40	0
	5	0	0	0	0	0	40

In conclusione dopo aver provato a trovare informazioni nei dati tramite le procedure di clustering è possibile affermare che si sono presentate alcune criticità. La prima consiste nella difficoltà di distinguere tra la classe "nana bruna" e "nana bianca", infatti solo un algoritmo su quattro è riuscito a dividere le due classi. Dai boxplot a pagina 6, è possibile vedere quanto le due classi siano vicine in termini di temperatura, luminosità relativa e raggio relativo.

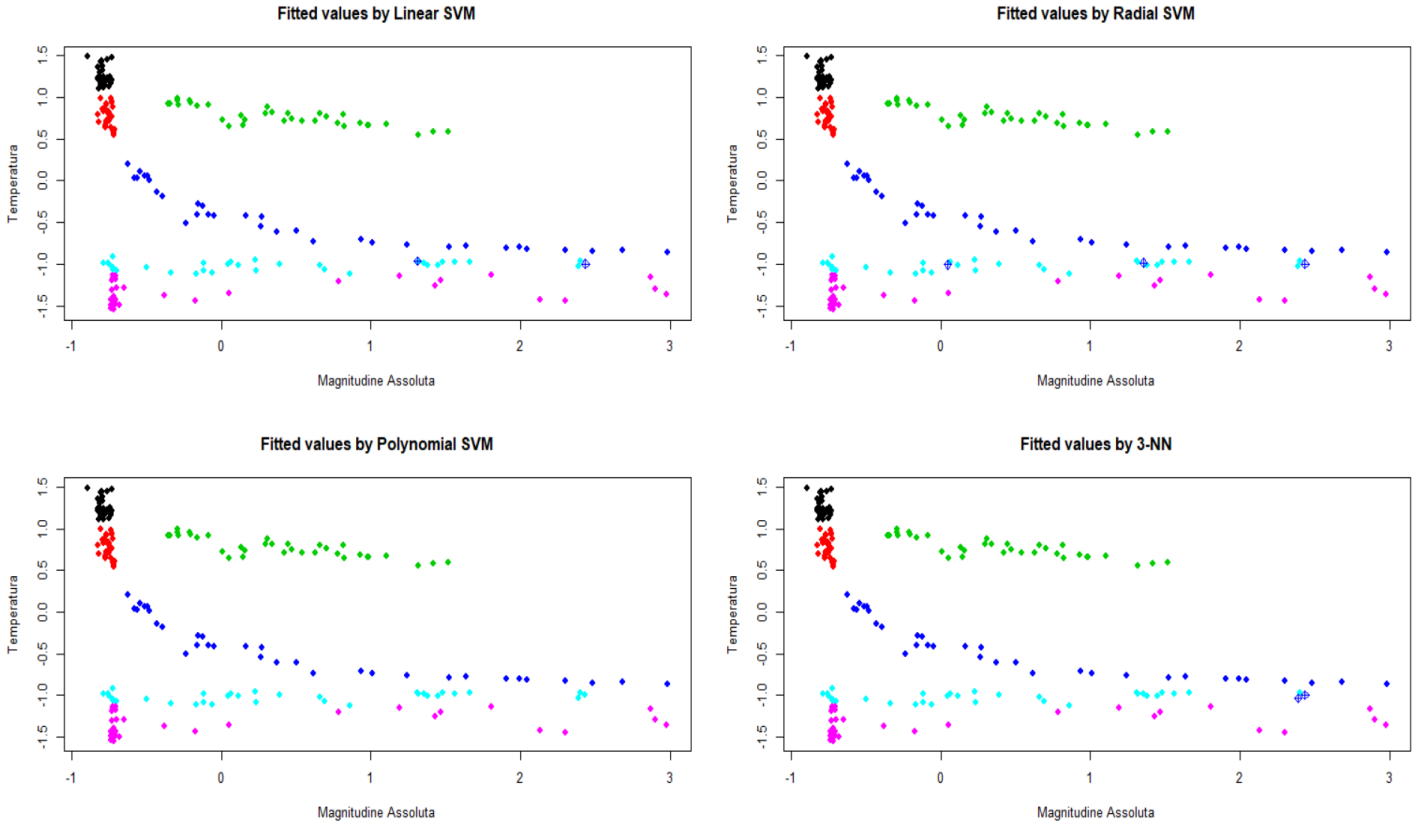
In secondo luogo è stata notata la difficoltà nell'individuare il cluster relativo alla classe "sequenza principale", la quale viene sempre in parte accorpata alle supergiganti e in parte alle nane bianche, questo perché per quanto riguarda la luminosità e la dimensione, è molto vicina alle stelle nane, e dal punto di vista della temperatura e della magnitudine assoluta si avvicinano molto di più alle supergiganti. Infine è evidente come le stelle che appartengono alla classe "ipergiganti" vengano riconosciute da tutti gli algoritmi utilizzati, sebbene al suo interno vengano divisi in due gruppi dal db-scan e dal k-means, infatti si differenziano dalle altre soprattutto in termini di dimensione.

Supervised Learning

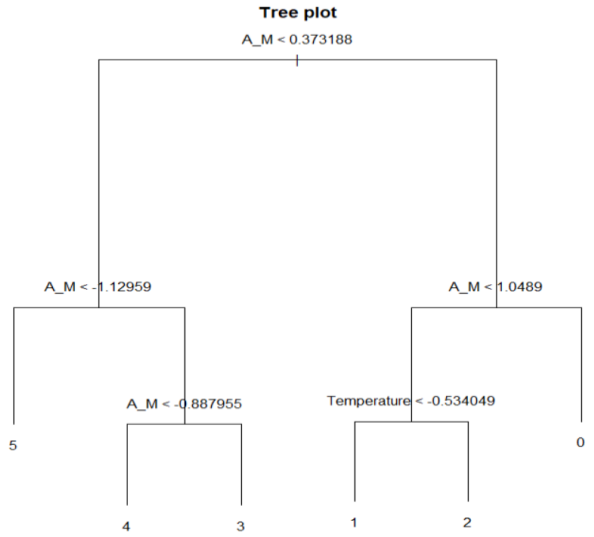
Il primo algoritmo supervisionato di classificazione considerato è il K-NN, che basa la sua forza di classificazione sulla distanza geometrica dei punti e attribuisce una nuova osservazione alla classe che contiene il numero maggiore di presenze trai k punti più vicini ad essa. Nel nostro lavoro è stato considerato un K-NN con iperparametro k pari a 1, 3, 5, 7 e 9 sul training set con cross validation leave one out. I numeri pari sono stati esclusi perché è dimostrabile che essi non possono massimizzare la performance dell'approccio di classificazione. La seguente figura evidenzia una buona accuracy, in particolare per k più piccoli; ma poiché k=1 potrebbe essere fonte di overfitting si preferirebbe nel caso un K-NN con k pari a 3 che assicura comunque soltanto 2 errori su 204 osservazioni di training set.

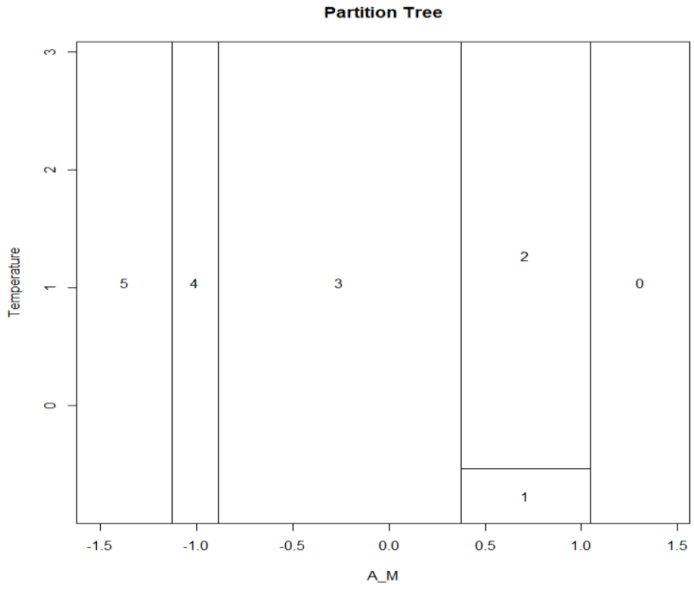


Successivamente è stato preso in considerazione l'approccio della Support Vector Machine nelle versioni lineare, polinomiale e kernel. Per quanto riguarda l'ottimizzazione degli iperparametri si è pensato di realizzare una griglia di possibili valori con lo scopo di minimizzare l'errore commesso sul validation set, 24 osservazioni estrapolate dal training set. Successivamente, trovate le migliori combinazioni degli iperparametri, si è proceduto alla cross validation leave one out sul training set completo per testarne la performance del modello. Per quanto riguarda la SVM lineare l'unico parametro da ottimizzare è C (sul software R è 'cost') che viene fissato pari a 1.53 fornendo accuracy pari a 1 sul training set ed errore di classificazione nullo sul validation set. Ma purtroppo mediante CV l-o-o sull'intero training set gli errori commessi da tale modello sono due; ciò ci porta a considerare SVM non lineari poiché evidentemente i dati non risultano linearmente separabili. Lo stesso procedimento viene applicato ai parametri da ottimizzare per la SVM con trasformazione kernel radiale, cost e gamma che rispettivamente risultano pari a 2.26 e 2.17. Tale soluzione commette un errore in più alla versione lineare dell'algoritmo (delle tre osservazioni classificate erroneamente una coincide con quelle della SVM lineare). Per un'ultima volta si procede all'ottimizzazione dei parametri, questa volta per il grado (degree), cost e gamma che riguardano la SVM polinomiale. La soluzione è un algoritmo che applica trasformazione quadratica (degree=2), cost pari a 1 e gamma pari a 0.8 che performa perfettamente anche nella leave one out cross validation del training set. In tal caso si hanno 61 Support Vectors nell'intero training set, ovvero 61 osservazioni che modificherebbero l'iperpiano non lineare che separa i dati se mancassero. A questo punto, più a scopo accademico che per una reale necessità, abbiamo considerato un algoritmo di auto machine learning al fine di trovare una migliore soluzione della tecnica e che andasse ad ottimizzare i parametri della SVM. Il risultato non è stato molto soddisfacente, poiché ne è conseguita una SVM lineare con parametro cost=1.32 con una performance largamente inferiore a quella della SVM polinomiale descritta poco fa. Di seguito si mostrano gli errori degli algoritmi sin qui specificati (due in SVM lineare, tre in SVM radiale, zero in SVM polinomiale, due in K-NN con K=3). Si evidenzia che tutti gli errori vengono commessi nella previsione della classe 'Ipergiganti' confusa con l'etichetta 'Sequenza Principale'.

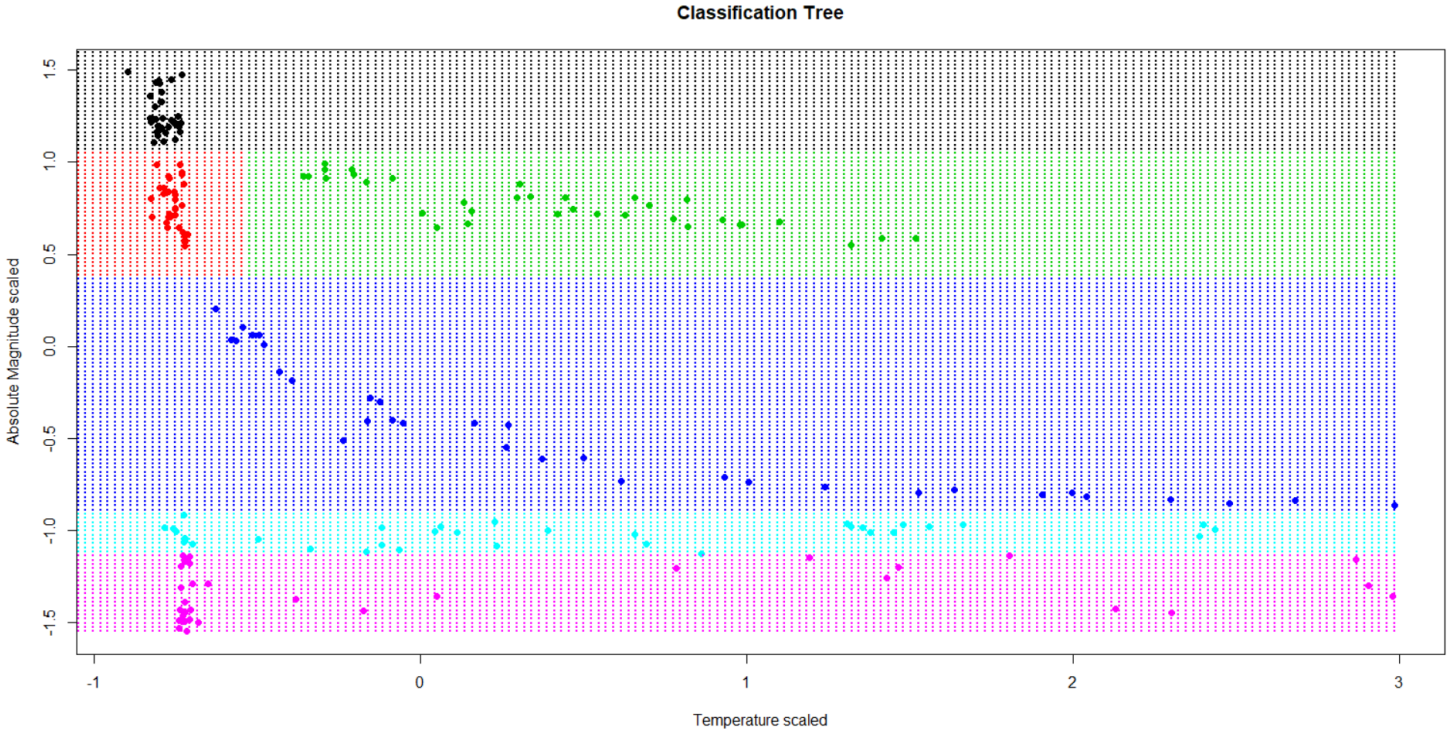


Nonostante potessimo ritenerci soddisfatti della performance (insuperabile in termini di accuracy e misclassification error rate) della SVM polinomiale con iperparametri degree=2, cost=1 e gamma=0.8, abbiamo considerato algoritmi basati sugli alberi di classificazione. In particolare è stato stimato un albero molto semplice che considera soltanto le due variabili più esplicative della risposta 'Type', Temperatura e Magnitudine Assoluta, e con parametri che gestiscono la fase di post pruning pari a: numero minimo di osservazioni di un potenziale nodo figlio uguale a 5, numero minimo di osservazioni di un potenziale nodo genitore uguale a 10 e devianza minima pari a 0.01. Sorprendentemente tale albero non commette alcun errore nel training set usato mediante cross validation leave one out nonostante sia composto soltanto da sei foglie, una per ciascuna classe della risposta, come si vede di seguito.



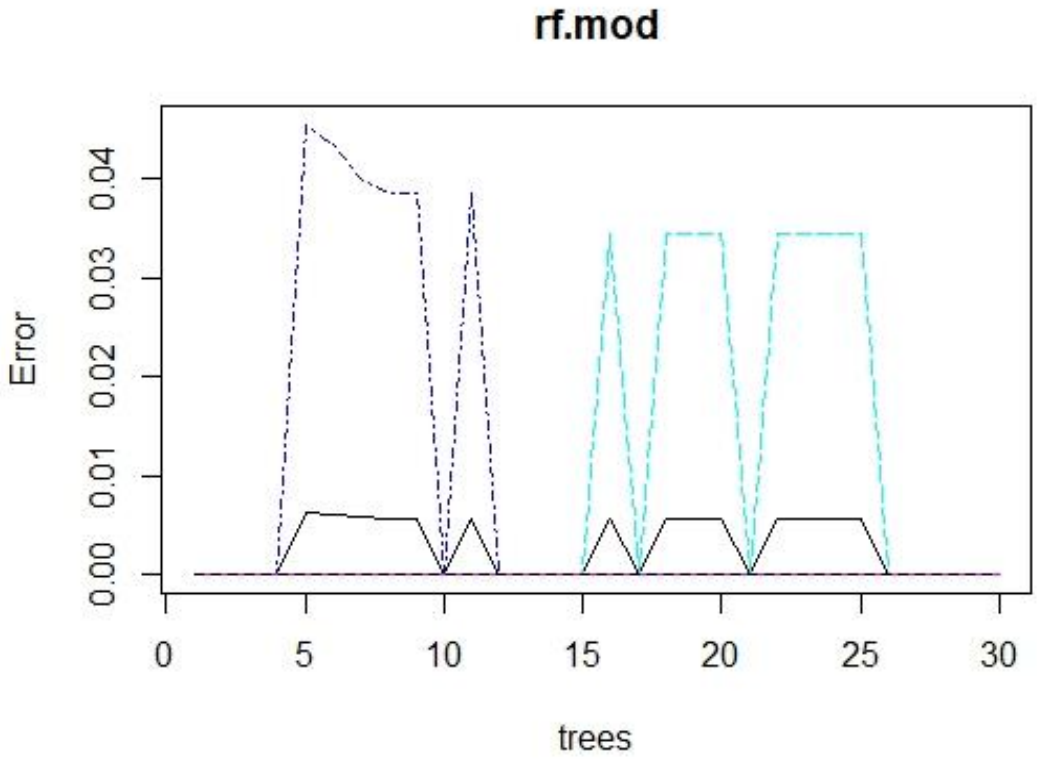


In effetti graficamente si nota che i dati sono facilmente separabili da un algoritmo che esegue dei 'tagli' nell'intervallo delle variabili esplicative. Allora lo spazio risulta suddiviso in sei zone con purezza massima come si nota sia dal partition tree plot sia di seguito.

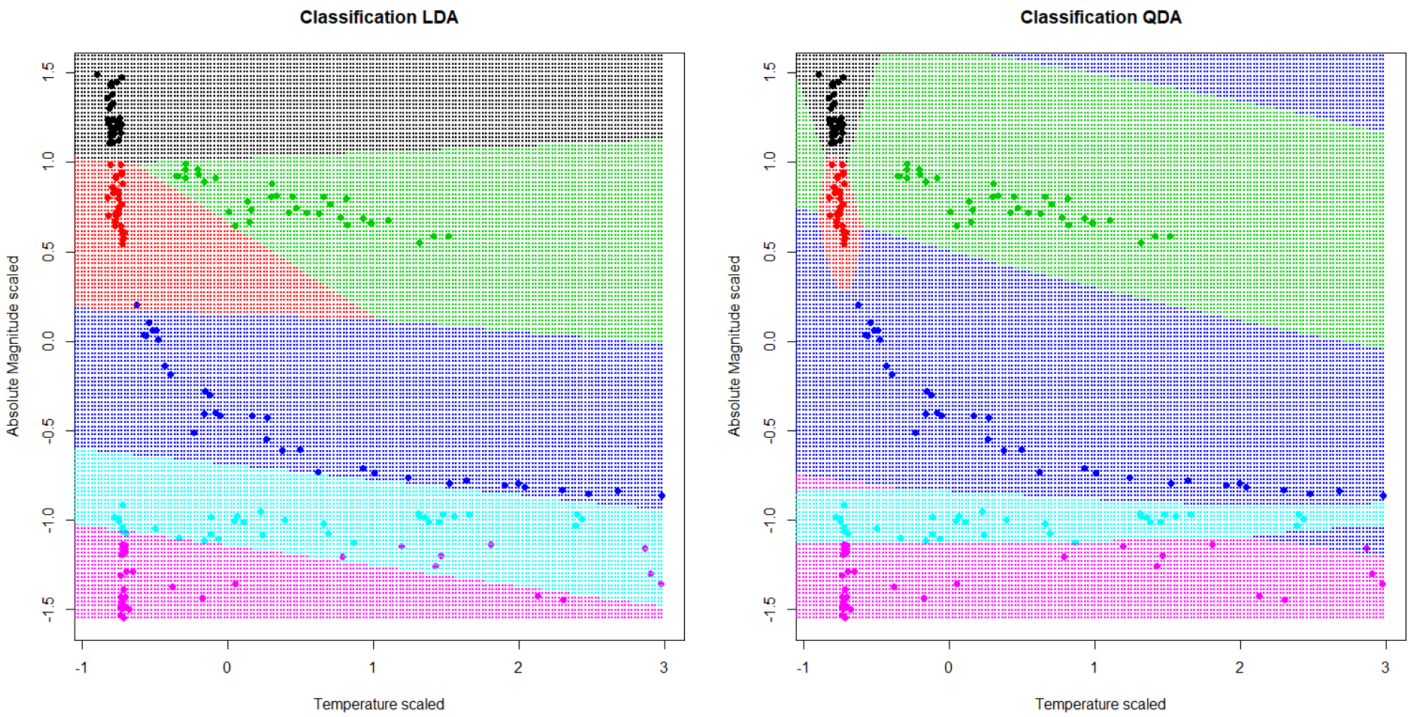


Siccome l'albero di classificazione specificato è decisamente idoneo al problema analizzato, si presuppone che anche la performance della Random Forest sia perfetta, infatti addestrandone uno che considera due variabili esplicative sulle quattro a disposizione in ciascuno dei trenta alberi che compongono la foresta appunto, l'accuracy e il misclassification error rate restano stabili su 1 e 0 rispettivamente. Questa volta, poiché la Random Forest al suo interno effettua ricampionamento bootstrap sulle osservazioni, l'accuracy è calcolata sul training set senza cross validation leave one

out e l'errore sullo stesso validation set considerato in precedenza. Il seguente grafico mostra la convergenza dell'errore a zero per ciascuna classe della risposta all'aumentare degli alberi di classificazione coinvolti.



Per completezza, sono stati considerati con le due variabili esplicative ampiamente utilizzate, Temperatura e Magnitudine Assoluta, anche i classificatori che sfruttano il teorema di Bayes per stimare le probabilità di appartenenza alle classi condizionate alle esplicative: Linear Discriminant Analysis e Quadratic Discriminant Analysis. Le loro performance sul training set sono mostrate dal seguente grafico che mette in luce le loro differenze legate alla stima della matrice di varianze e covarianze delle variabili prese in considerazione. Evidentemente e prevedibilmente LDA ha una regola di classificazione più rigida di quella quadratica che non consente la corretta classificazione di parecchie osservazioni presenti nel training set, a differenza dell'approccio più dispendioso computazionalmente che invece ha un missclassification error pari ad uno nel set di dati di allenamento. Ad ogni modo nessuno dei due sarà scelto come modello finale.



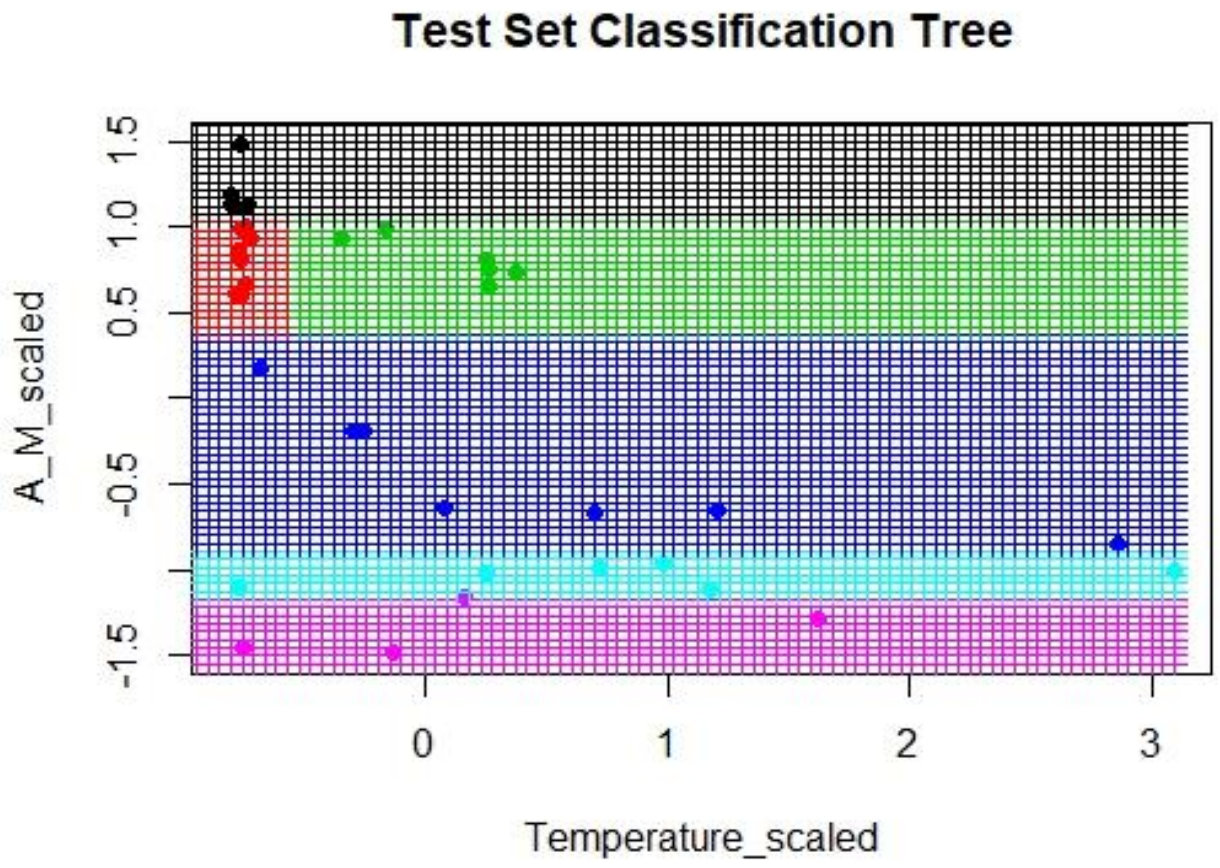
In conclusione è possibile fare una considerazione di tipo strutturale sui dati riferiti alle stelle analizzate: mediante SVM le osservazioni non risultano linearmente separabili rispetto alla propria classe di appartenenza, ma considerando un albero di classificazione sulle variabili esplicative Temperatura e Magnitudine assoluta, risultano sufficienti i 'tagli' dell'algoritmo (lineari per l'appunto). Ciò evidenzia una particolare distinzione tra le stelle e quindi una non eccessiva difficoltà nel classificarle. Risulta dunque evidente che le osservazioni non sono sovrapposte con etichette di classe diverse considerando anche soltanto tali variabili. Per quanto riguarda la Support Vector Machine solo la trasformazione polinomiale (quadratica) consente la perfetta classificazione delle osservazioni a disposizione. Invece l'algoritmo KNN non è deludente ma non riesce a fornire l'accuracy perfetta degli altri due algoritmi suddetti a causa delle osservazioni che si trovano al limite tra le classi 'Nana bianca' e 'Sequenza principale'. In generale queste due classi hanno rappresentato la maggior difficoltà nell'uso dell'approccio di Supervised Learning.

Discussioni

Spinti dall'interesse per l'astronomia e visti i progressi teorici degli ultimi decenni nel campo della classificazione stellare, il presente elaborato si pone l'obiettivo di specificare un modello capace di identificare la tipologia di stella date le caratteristiche riguardanti temperatura, luminosità, raggio e magnitudine assoluta. Dopo aver esplorato i dati in maniera minuziosa e compreso il problema ci si è cimentati nell'apprendimento non supervisionato e supervisionato ottenendo ottimi risultati di interpretabilità. Nello specifico, dalle tecniche di clustering sono emerse difficoltà nel distinguere tra la classe "nana bruna" e "nana bianca", infatti solo un algoritmo su quattro è riuscito a dividere le due classi, l'algoritmo EM. Inoltre si è notato che individuare il cluster relativo alla classe "sequenza principale" non è stato quasi mai possibile perché, essendo la classe di transizione delle stelle, viene spesso scambiata con la classe delle supergiganti, in quanto in termini di temperatura e magnitudine

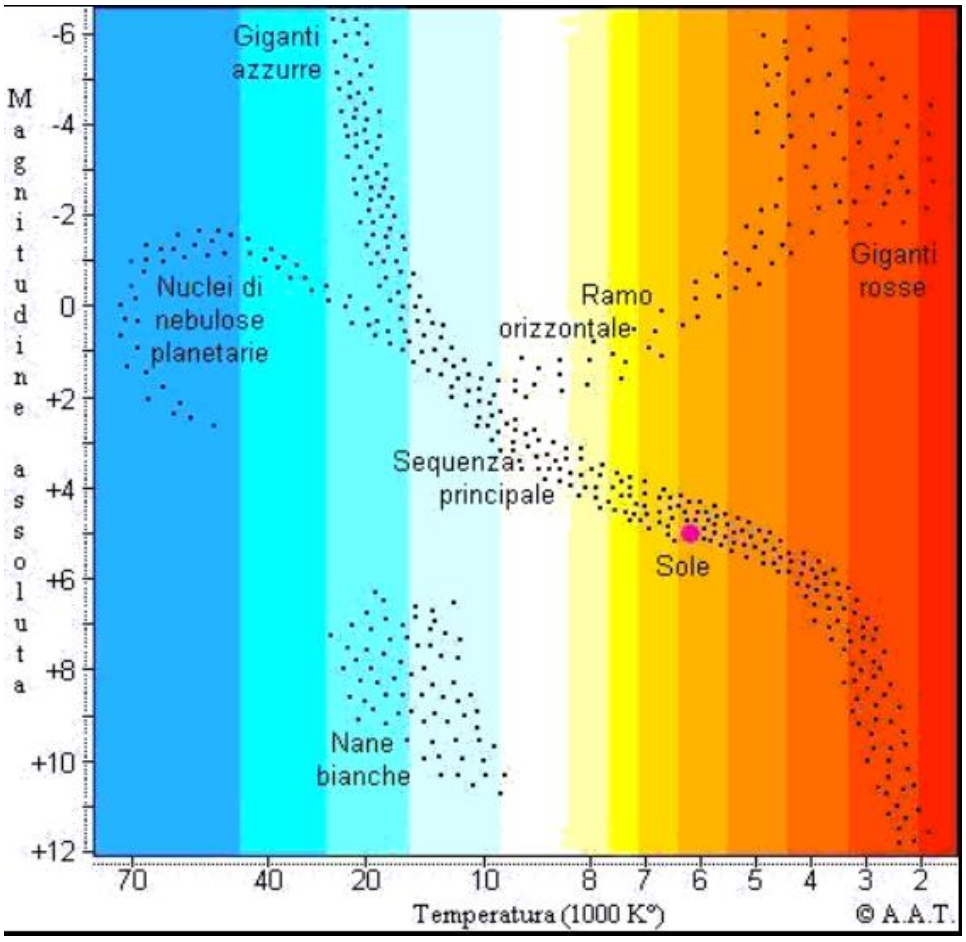
assoluta non ci sono grandi differenze, e talvolta confusa anche alla classe delle nane bianche, perché per quanto riguarda la luminosità e la dimensione, queste due classi sono molto simili. Infine è evidente come le stelle che appartengono alla classe "ipergiganti" vengano riconosciute da tutti gli algoritmi utilizzati, sebbene al suo interno vengano divisi in due gruppi dal db-scan e dal k-means, infatti si differenziano da tutti gli altri tipi di stelle soprattutto per la dimensione.

Invece per quanto riguarda il supervised learning, si è proceduto sia con algoritmi basati sulla distanza geometrica dei punti, in primis il KNN (ottimale con k=3 per evitare il possibile overfitting dell'algoritmo con k=1) ingannato però dalla vicinanza tra le stelle della sequenza principale e le ipergiganti, errando due stelle del training set, sia con algoritmi di natura probabilistica, con Support Vector Machine nelle versioni lineare, con kernel radiale e polinomiale, ottenendo un'accuracy perfetta con l'ultima delle tre e iperparametri grado=2, C pari a 1 e gamma pari a 0.8, con albero di classificazione e Random Forest (già il primo ottiene la massima performance) e infine con Linear Discriminant Analysis e Quadratic Discriminant Analysis. Come modello finale per fare previsione sul test set viene selezionato l'albero di classificazione sulle variabili Temperatura e Magnitudine Assoluta essendo il modello più semplice e con la massima accuracy raggiunta sul training set con Cross Validation Leave One Out tra quelli stimati. La classificazione è ancora una volta perfetta come si nota nel seguente grafico.



A tale conclusione ottenuta dal nostro modello ci colleghiamo con un importante risultato teorico raggiunto da due fisici e astrologi che hanno studiato il fenomeno lo scorso secolo: Hertzprung e

Russell. Essi hanno infatti introdotto il diagramma che porta il loro nome, il quale è uno strumento teorico che mette in relazione la temperatura efficace (riportata in ascissa) e la luminosità (riportata in ordinata) delle stelle. Le due grandezze sono quantità fisiche che dipendono strettamente dalle caratteristiche intrinseche della stella (massa, età e composizione chimica), non sono misurabili direttamente dall'osservatore, ma possono essere derivate attraverso modelli fisici. Il diagramma H-R viene utilizzato per comprendere l'evoluzione stellare e le caratteristiche fisiche delle singole stelle e degli agglomerati stellari al fine di determinare l'età, la composizione chimica e la distanza di una popolazione stellare. Lo riportiamo di seguito, dove viene evidenziata anche la posizione del Sole secondo tali caratteristiche.



Da un primo esame del diagramma H-R si osserva immediatamente come le stelle tendano a posizionarsi in regioni ben distinte: la struttura evolutiva predominante è la diagonale che parte dall'angolo in alto a sinistra (dove si trovano le stelle più massicce, calde e luminose) verso l'angolo in basso a destra (dove si posizionano le stelle meno massicce, più fredde e meno luminose), chiamata la sequenza principale di età zero. In basso a sinistra si trova la sequenza delle nane bianche, in quanto sono corpi celesti con alte temperature efficaci ma poco luminose; mentre sopra la sequenza principale, verso destra, si dispongono le giganti rosse e le supergiganti, in quanto sono corpi celesti molto luminosi ma a basse temperature.

Siamo dunque molto soddisfatti di aver fornito un risultato in linea con gli studi teorici del settore.