Starclassification

federico.dalba

May 2021

Indice

1 Data understanding

Il set di dati a nostra disposizione ha dimensione 240x7 e la nostra variabile target è l'attributo Type. Questo assume valori tra un range 0 e 5 a cui sono associati il tipo di stella: red dwarf, brown dwarf, white dwarf, main sequence, super giants e hyper giants. La tipologia della stella è distribuita uniformemente nei nostri dati come si può vedere in Fig. 1.

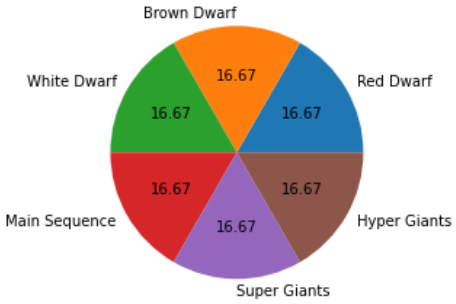


Figura 1: *Distribuzione della target class nel nostro dataset*

1.1 Temperature

La temperatura delle stelle nel nostro dataset assume un valore minimo di 1939 K e una massima di 40000 K. Il 50% delle stelle ha una temperatura inferiore a 5776 K ed inoltre la distribuzione alle varie temperature non sembra dipendere dalla tipologia della stella, come si può vedere in Fig. 2

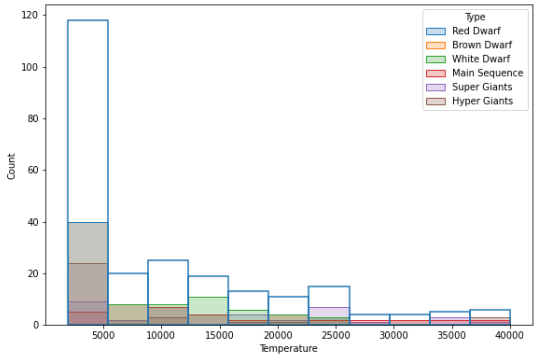


Figura 2: *Distribuzione della temperatura nel nostro dataset*

1.2 Relative luminosity

L'attributo L è la luminosità relativa, cioè il rapporto tra luminosità misurata e luminosità media del Sole. Il valore più piccolo assunto nel nostro dataset per la luminosità è 0.000080 mentre quello massimmo è 849420 e dunque abbiamo un attributo che attraversa svariati ordini di grandezza. Il 50% dei valori della luminosità sono inferiori a 0.0705 ed inoltre la tipologia di stella sembra influire sul valore assunto da L, infatti per esempio soltanto le stelle del tipo Super giants o iper giants hanno un valore della luminosità maggiore di 700000 come possiamo vedere in Fig. 3

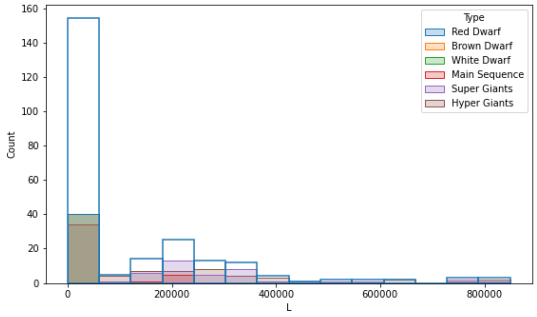


Figura 3: *Distribuzione della luminosità relativa nel nostro dataset*

1.3 Radius

R è il rapporto tra il raggio effettivo della stella e il raggio del Sole. Il valore minimo assunto nel dataset per questo attributo è 0.0084 mentre il valore massimo è 1948.5. Anche per questo attributo la distribuzione è asimmetrica, infatti il 50% delle stelle nel nostro dataset ha il raggio minore di 0.7 volte quello del Sole. Dalla Fig 4 si può vedere che la maggior parte di stelle si trovano nella prima bar dell'istogramma e le bar successive sono i valori delle stelle Super giants e iper giants.

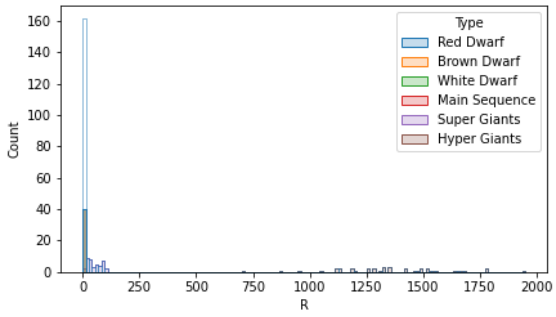


Figura 4: *Distribuzione del raggio relativo nel nostro dataset*

Per lavorare meglio con questo attributo ho deciso di trasformarlo usando il logaritmo in base dieci in modo da avere una distribuzione meno 'sparsa'.

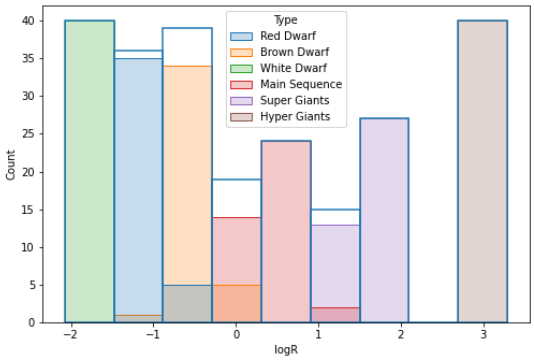


Figura 5: *Distribuzione del logaritmo in base 10 del raggio relativo nel nostro dataset*

1.4 Absolute magnitude

La magnitudine assoluta è misurata in scala logaritmica inversa e rappresenta la magnitudine apparente che avrebbe una stella se fosse a 10 parsec di distanza.

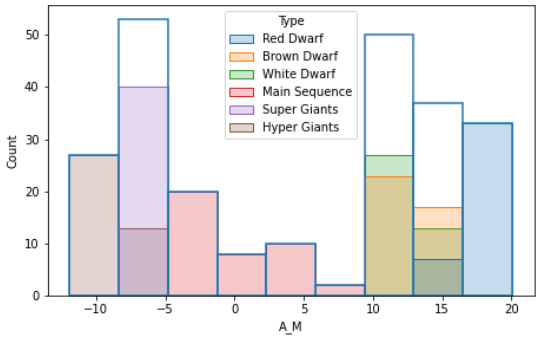


Figura 6: *Distribuzione della magnitudine assoluta nel nostro dataset. Possiamo vedere che c'è una distinzione quasi netta tra i tipi di stelle.*

1.5 Color and Spectral Class

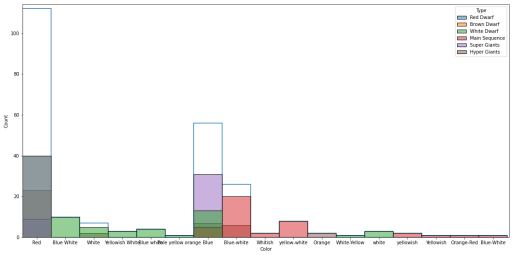


Figura 7: *Distribuzione del colore delle stelle*

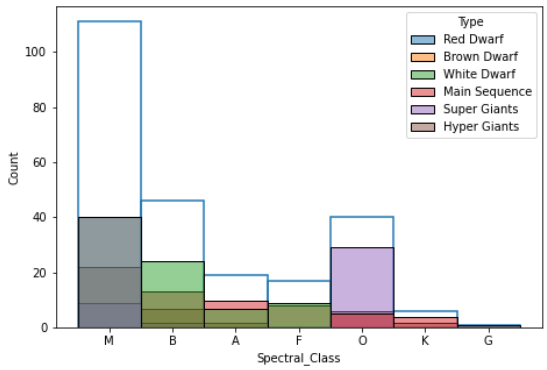


Figura 8: *La classe spettrale viene assegnata in base allo spettro di emissione della stella e al colore.*

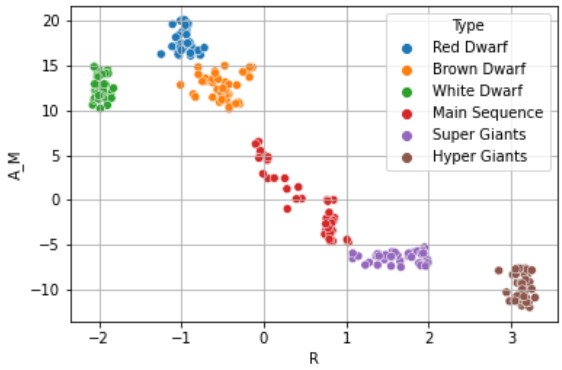
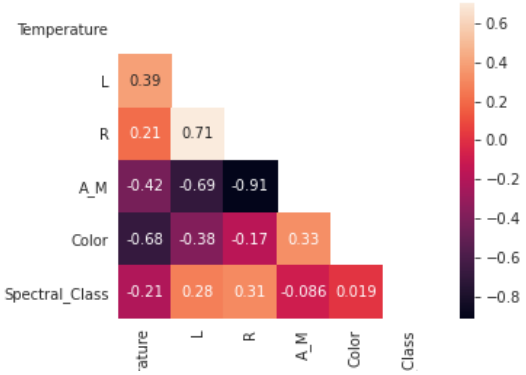


Figura 9: *Scatterplot tra magnitudine assoluta e logR.*

1.6 Correlations between attributes



Dalla matrice di correlazione vediamo che:

- La magnitudine assoluta e il logaritmo del raggio sono fortemente anticorrelate(Vedi Fig. 9)
- Il logaritmo del raggio è debolmente correlato con la luminosità
- Il colore e la temperatura sono debolmente anticorrelate

Inoltre

- Il logaritmo della luminosità è fortemente anticorrelato con la magnitudine assoluta come si può vedere in Fig. 10

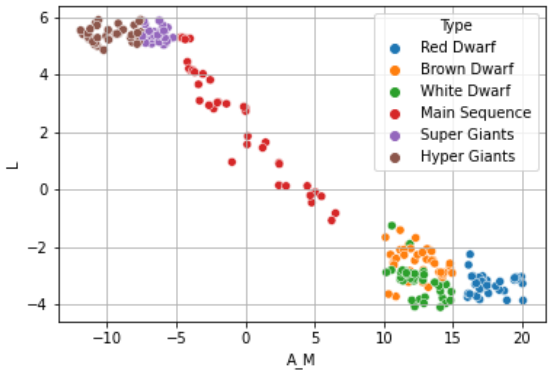


Figura 10: *Scatterplot tra logaritmo della luminosità relativa e magnitudine assoluta. Dalla figura vediamo che sono fortemente anticorrelate*

Inoltre il plot tra magnitudine assoluta e temperatura ci rende il diagramma di Hertzsprung-Russel.

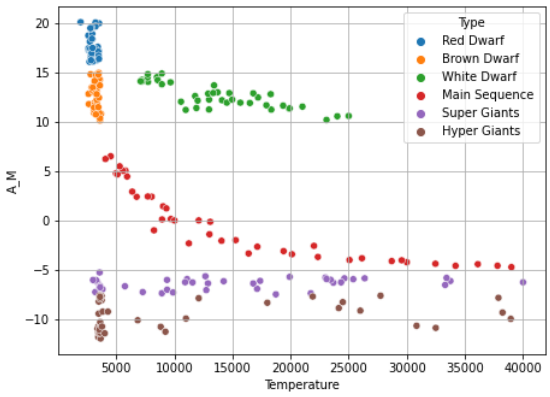


Figura 11: *Diagramma di Hertzsprung-Russel*

2 Supervised learning

Lo scopo di questa sezione è quella di trovare il classificatore che mi rende le migliori performance per questo dataset. Il modus operandi è stato all'incirca lo stesso per tutti i classificatori: Ho eseguito una data reduction, normalizzato i dati , ho cercato gli iperparametri migliori e dopodichè ho valutato le performance del classificatore utilizzando la crossvalidation. Per la ricerca degli iperparametri ho usato la funzione randomsearchCV(o gridsearchCV) di sklearn sul mio spazio di parametri e ho selezionato i valori per cui lo scoring 'accuracy' fosse massimizzato. Per quanto riguarda la data reduction ho agito in tre modi diversi:

- Ho tolto gli attributi a mano, togliendo via via quelli che avevano un coefficiente di correlazione più alto con uno degli altri attributi
- utilizzando la principal component analysis(PCA).
- Utilizzando la feature importance del RandomForest classifier

Ho poi eseguito un confronto tra questi tre metodi comparando dataset con la stessa dimensione. L'attributo L invece è stato rimosso perchè ridondante.

2.1 Decisiontree classifier

I decisiontree classifier non hanno bisogno di data reduction. L'algoritmo di ricerca degli iperparametri per questo classificatore ci ha selezionato:

$$\text{min\_samples\_split} = 10 \qquad \text{min\_samples\_leaf} = 1$$

$$\text{max\_depth} = 10 \qquad \text{criterio} :' \text{gini}'$$

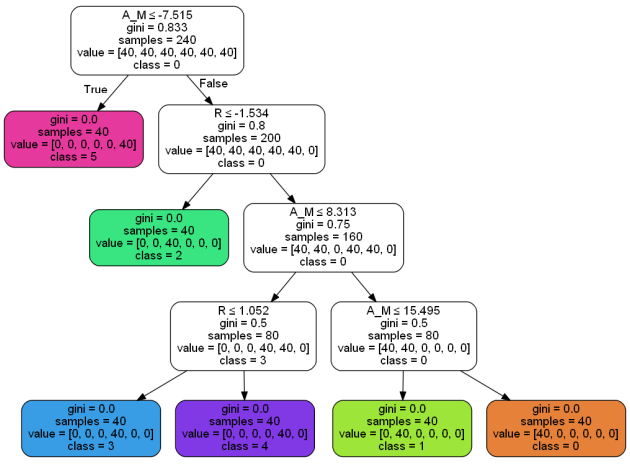


Figura 12: *Rappresentazione dell'albero ottenuto dal nostro dataset usando randomsearchcv*

Con questi valori possiamo vedere una rappresentazione del nostro albero in Fig. 12

Per quanto riguarda le performance, l'accuratezza utilizzando la crossvalidation è:

$$\text{ACC}_{\text{decisiontree}} = 0.99 \pm 0.02$$

2.2 Randomforest classifier

Il randomforest classifier crea un ensamble di decision tree. Le performance sul dataset sono le stesse ottenute con il decision tree:

$$\text{ACC}_{\text{RandomForest}} = 0.99 \pm 0.02$$

Questo classificatore, tuttavia, ci permette anche di eseguire una feature selection in base a quanto ciascun attributo influisce sull'information gain ad ogni splitting.(Fig.13)

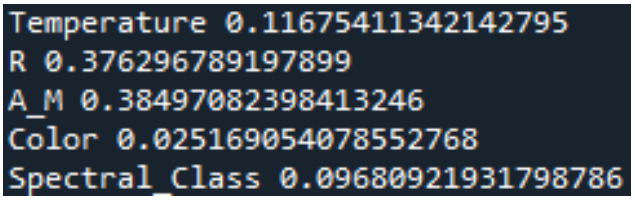


Figura 13: *Feature importance.*

2.3 KNN classifier

Il KNN performa male per dataset di dimensioni troppo grandi. Quindi ho selezionato datasets di dimensione 2 e 3.

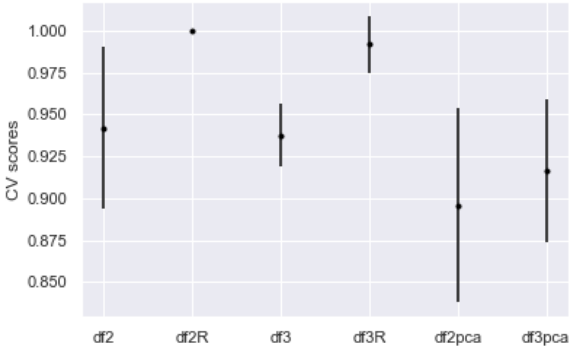


Figura 14: *Cross validation mean accuracies al variare dei datasets. Il numero accanto a df rappresenta la dimensione del dataset. L'aggiunta della R rappresenta la feature selection eseguita con Random Forest, mentre pca la feature selection tramite principal component analysis*

Dalla Fig. 14 vediamo che i peggiori risultati li otteniamo selezionando gli attributi tramite pca. I migliori risultati li otteniamo riducendo la dimensione del dataset tramite feature selection di randomforest. In

particolare, il miglior valore lo otteniamo per il set bidimensionale e per un valore dell'iperparametro ottimale $k_{df2R} = 4$ da cui otteniamo un valore dell'accuracy(per il dataset tridimensionale)

$$ACC_{KNN} = 1.0 \pm 0.0$$

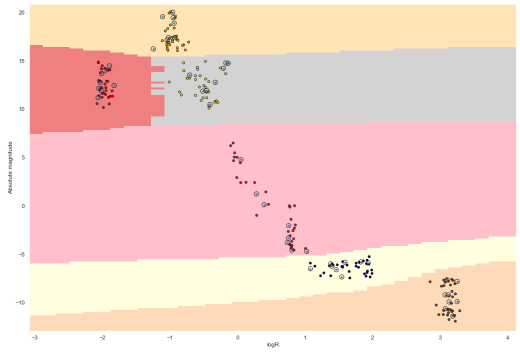


Figura 15: *Rappresentazione del classificatore KNN per il dataset bidimensionale. I punti cerchiati fanno parte del dataset e verranno classificati a seconda del colore in cui si trovano.*

2.4 SVM

Per questo classificatore ho utilizzato gridsearch per cercare i miglior valori dei parametri di C e γ ed il miglior kernel tra polinomiale, lineare e rbf.

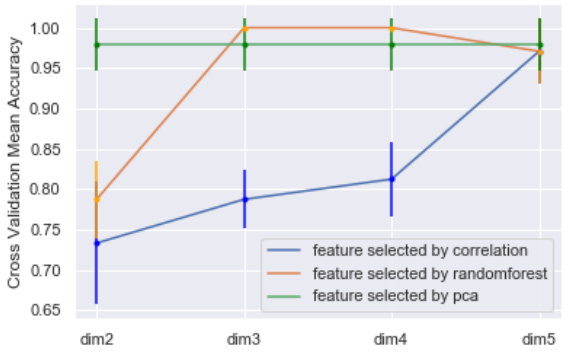


Figura 16: *Accuratezza del classificatore a variare delle dimensioni del test set.*

Dalla Fig. 16 vediamo che il risultato migliore lo otteniamo in dimensione 3(Temperatura, logR e AM) con gli attributi selezionati tramite randomforest classifier. Gli iperparametri migliori trovati sono: $C = 10$, $\gamma = 0.1$, kernel = *linear* per cui otteniamo un valore dell'accuratezza media:

$$ACC_{SVM} = 1.0 \pm 0.0$$

2.5 NN classifier

Data la poca quantità di records, per ottenere i migliori risultati con questo classificatore ho cercato di tenere le dimensioni del network il più piccole possibile e quindi ho direttamente utilizzato un dataset di dimensione ridotta(feature selezionate con random forest: logR, Temperatura e AM). Ho definito due modelli: uno con con singolo strato ed uno con due strati, dopodichè ho avviato il processo di training utilizzando la 3-fold cross validation e calcolato l'accuratezza media.

$$ACC_{1layer} = 0.98 \pm 0.02$$

$$ACC_{2layers} = 0.99 \pm 0.01$$

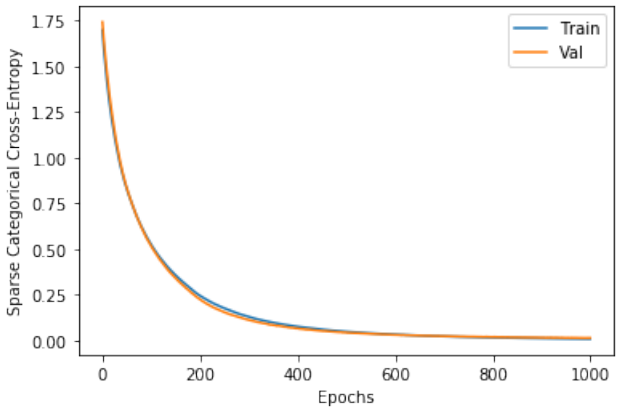


Figura 17: *Validation loss e training loss nel modello con singolo layer. Training set e validation set per una fold della cross validation*

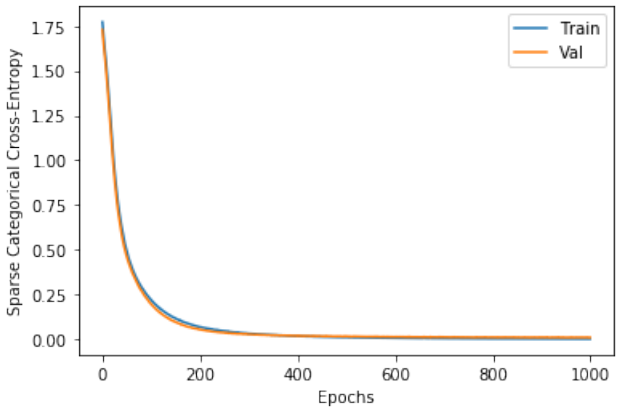


Figura 18: *Validation loss e training loss nel modello con due layers. Training set e validation set per una fold della cross validation*

Ho deciso di mettere per ciascun layer un numero di neuroni poco più grosso della input dimension, questo perchè per network troppo complessi abbiamo overfitting come possiamo vedere in Fig. 19.

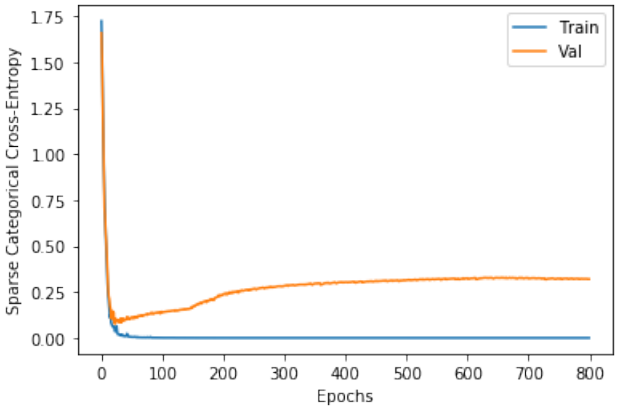


Figura 19: *Validation loss per un deep model con 64 nodi per hidden layer.*

Questo risultato si può migliorare aggiungendo un dropout, tuttavia il risultato ottenuto non è migliore di quello con i modelli a 1 o 2 strati.

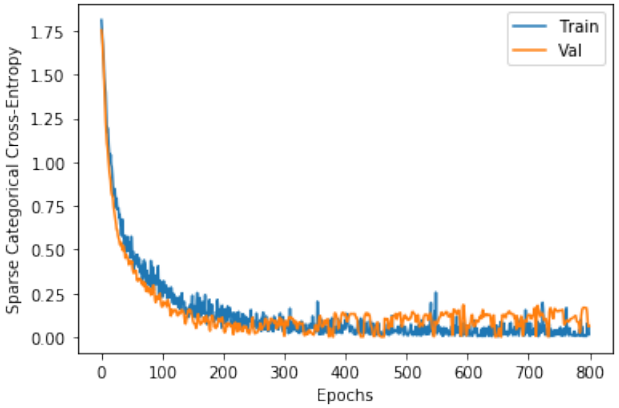


Figura 20: *Aggiungendo un dropout miglioriamo il risultato del deep model.*

3 Conclusioni

Tutti i classificatori utilizzati hanno delle buone performance per questo problema di classificazione. Il classificatore SVM mostra le migliori performance(nessuna missclassificazione neanche utilizzando la cross validation) in un dataset tridimensionale, mentre il classificatore KNN mostra le migliori performance per una dataset bidimensionale. L'utilizzo della pca per ridurre la dimensione del dataset mostra sempre ottime performance a qualsiasi dimensione, tuttavia i migliori risultati sono stati ottenuti riducendo la dimensione tramite feature selection del random forest classifier. Quest'ultimo mostra ottime performance anche utilizzando l'intero dataset ed inoltre ci permette di selezionare gli attributi più importanti, che per questo dataset sono : Magnitudine assoluta, logaritmo del raggio e temperatura. Per quanto riguarda le Neural Network i modelli ad uno e due strati mostrano delle buone performance e dalle Fig. 17 e Fig. 18 vediamo che il learning rate è abbastanza decente anche se un pò rapido per il modello a due strati.