Progetto di Machine Learning

Appello del 17/02/2020

Stellar Classification

Christian Bernasconi 816423
Riccardo Pozzi 807857
Marco Ripamonti 806785

Indice

1 Introduzione

L'obiettivo di questo elaborato è l'analisi ed elaborazione di un dataset per determinare un buon classificatore, costruito con modelli di machine learning, sulla base di varie misure di performance.
Il dataset scelto ed individuato sulla base dell'interesse comune, contiene dati ottenuti dall'osservazione di stelle rispetto, ad esempio, luminosità o grandezza ed altre loro caratteristiche.
L'obiettivo è quindi quello di classificare una stella esaminando nel loro complesso gli attributi ad esse associati.

La scelta è ricaduta su questo dataset anche per poter avere la possibilità di applicare sia algoritmi che utilizzano esclusivamente features numeriche che algoritmi che coinvolgono features categoriche, per avere infine la possibilità di confrontarli tra loro.

2 Dataset

Il dataset[1] è composto da un insieme di 240 record e ognuno di questi è descritto da 7 features di cui 4 numeriche e 3 categoriche. Per rendere più veloce ed intuitivo il loro utilizzo, le features sono state rinominate in Temp, Lum, Rad, AbsMagn, Color, SpectrClass e Type.

- Temp: misura della temperatura sulla superficie di una stella (gradi Kelvin); numerica
- Lum: misura della luminosità di una stella calcolata rispetto alla luminosità del sole; numerica
- Rad: misura del raggio di una stella calcolata rispetto al raggio del sole; numerica
- AbsMagn: misura della magnitudine assoluta di una stella; numerico
- Color: colore di una stella; categorica (19 livelli)
- SpectrClass: classe spettrale di una stella; la classificazione adottata in questo dataset è la classificazione Harvard; categorica (7 livelli)

[1] Link al dataset: https://www.kaggle.com/deepu1109/star-dataset

- Type: tipo di una stella; feature adottata come target per la classificazione di una stella; categorica (6 livelli)

I valori di *Type* sono stati ricodificati da numeri a stringhe per facilitare la futura interpretazione dei risultati ottenuti. Di seguito viene indicata la codifica:

- 0 → BrownDwarf
- 1 → RedDwarf
- 2 → WhiteDwarf
- 3 → MainSequence
- 4 → Supergiant
- 5 → Hypergiant

I valori numerici all'interno del dataset sono di ordini di grandezza anche molto differenti, si è quindi eseguito uno scaling per uniformarli.
I valori della feature *Color* sono risultati piuttosto "sporchi". Alcuni di essi si ripresentano scritti con formattazioni differenti. Inizialmente c'è stato un tentativo di uniformare i colori equivalenti fin dove possibile (es: "White-Yellow" = "yellow-white", "Blue white" = "Blue-white") riducendo i levels da 19 a 11. Dopo uno studio sul dominio di interesse, però, si è notato che esistono due diverse scale di colore associate ad una stella: "vega-relative chromaticity" e "chromaticity" [2]. Non sapendo quindi a quale categoria i nostri dati facessero riferimento e dallo studio successivo della correlazione tra variabili (sezione 3.2), consideriamo non molto affidabile la feature Color, in quanto potrebbe portare a valutazioni errate.

3 Analisi esplorativa

3.1 Distribuzione target

Per prima cosa è stata verificata la distribuzione del target tra le varie istanze del dataset. Come mostrato in *Figura 1* e in *Tabella 1*, si può notare che il dataset è perfettamente bilanciato, in quanto conta 40 istanze per ognuna delle 6 classi.

[2] "Stellar classification - Wikipedia."
https://en.wikipedia.org/wiki/Stellar_classification#Harvard_spectral_classification

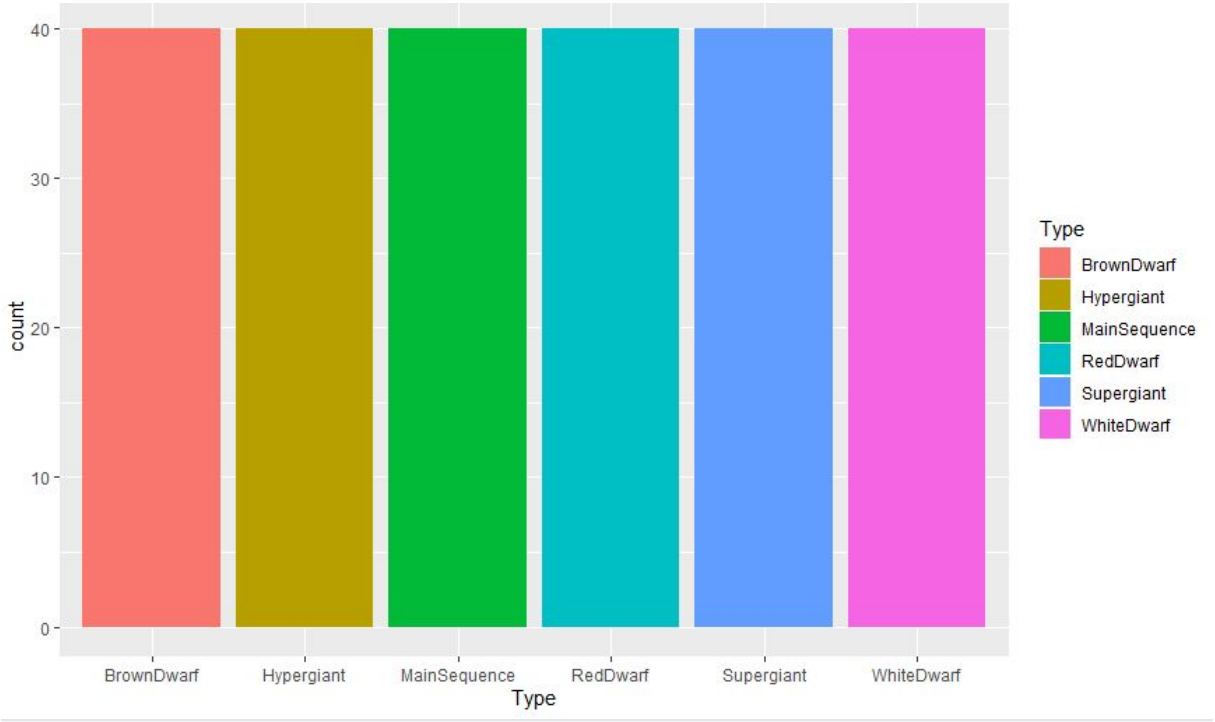

***Figura 1**: distribuzione del target*

Type	Distribuzione
BrownDwarf	40
Hypergiant	40
Mainsequence	40
RedDwarf	40
Supergiant	40
WhiteDwarf	40

***Tabella 1**: distribuzione del target*

3.2 Analisi features

Per verificare eventuali legami tra le features numeriche è stata calcolata la matrice di correlazione mostrata in *Figura 2*. Come si può osservare, nessuna coppia di variabili presenta un grado di correlazione estremamente vicino a 1 in valore assoluto. Visto il basso numero di features e data la mancanza di forti correlazioni, non si è ritenuta necessaria alcuna riduzione della dimensionalità.

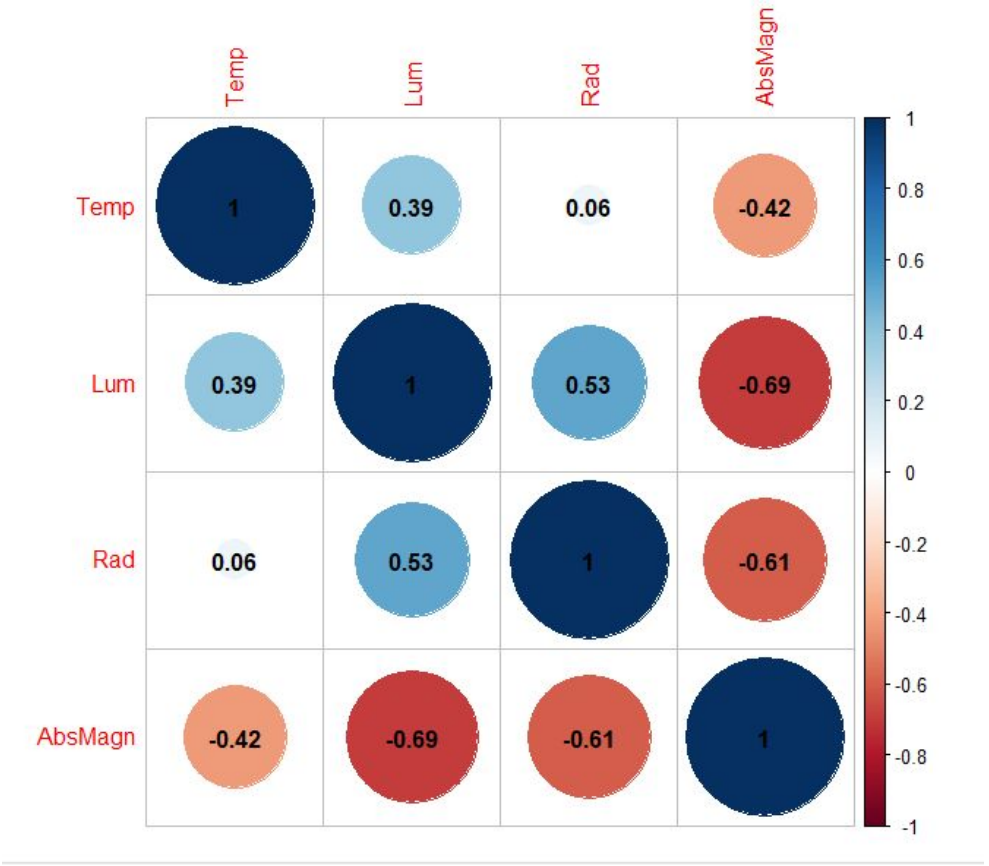

***Figura 2**: Matrice di correlazione features numeriche*

Continuando con le considerazione sulle features numeriche, è stata verificata la separazione delle classi per ogni coppia di features attraverso il grafico in *Figura 3*. Dal grafico si può notare che la feature Rad distingue molto bene il target Hypergiant da tutti gli altri, e che inoltre la migliore separazione sembra essere data dalla coppia Temp e AbsMagn come si vede in dettaglio in *Figura 4*.

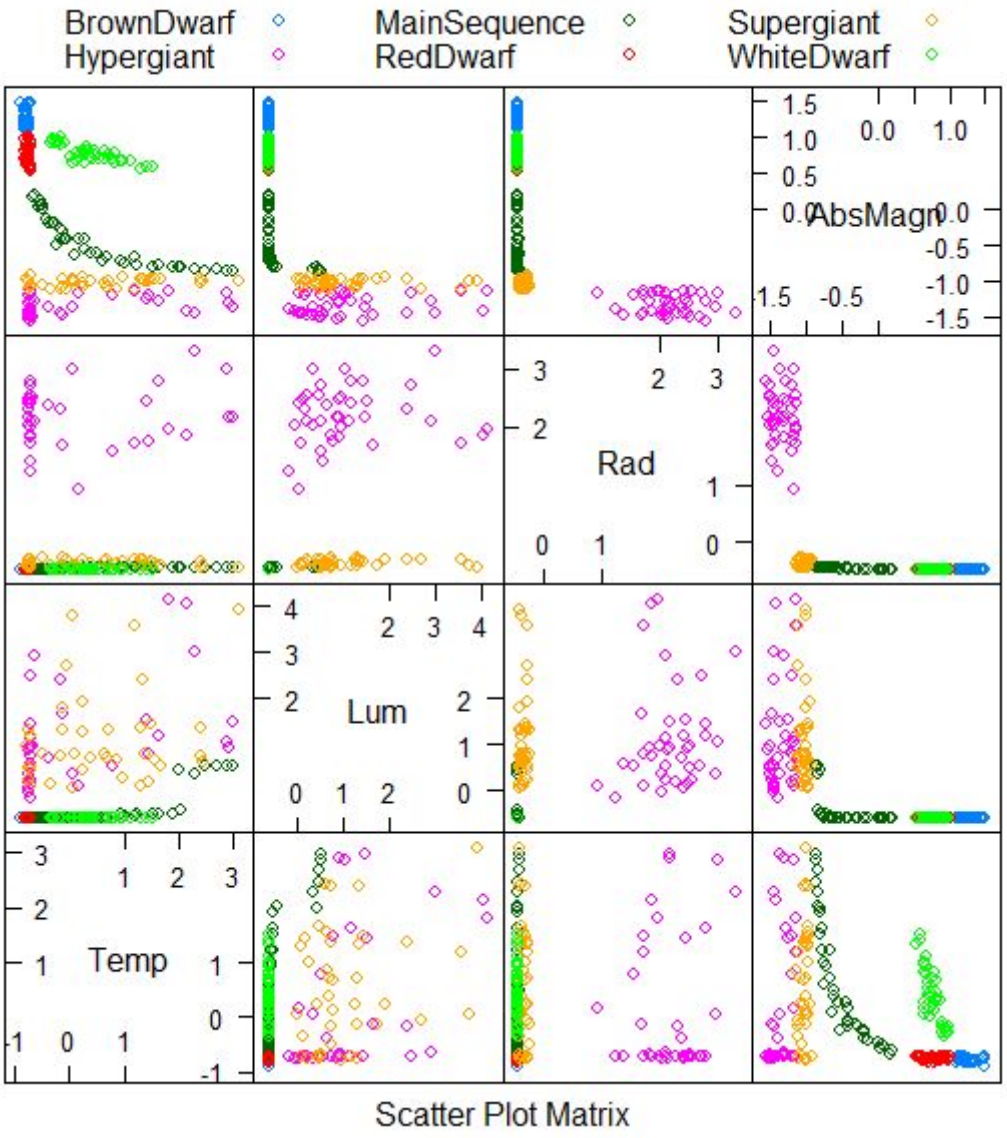

***Figura 3**: features plot*

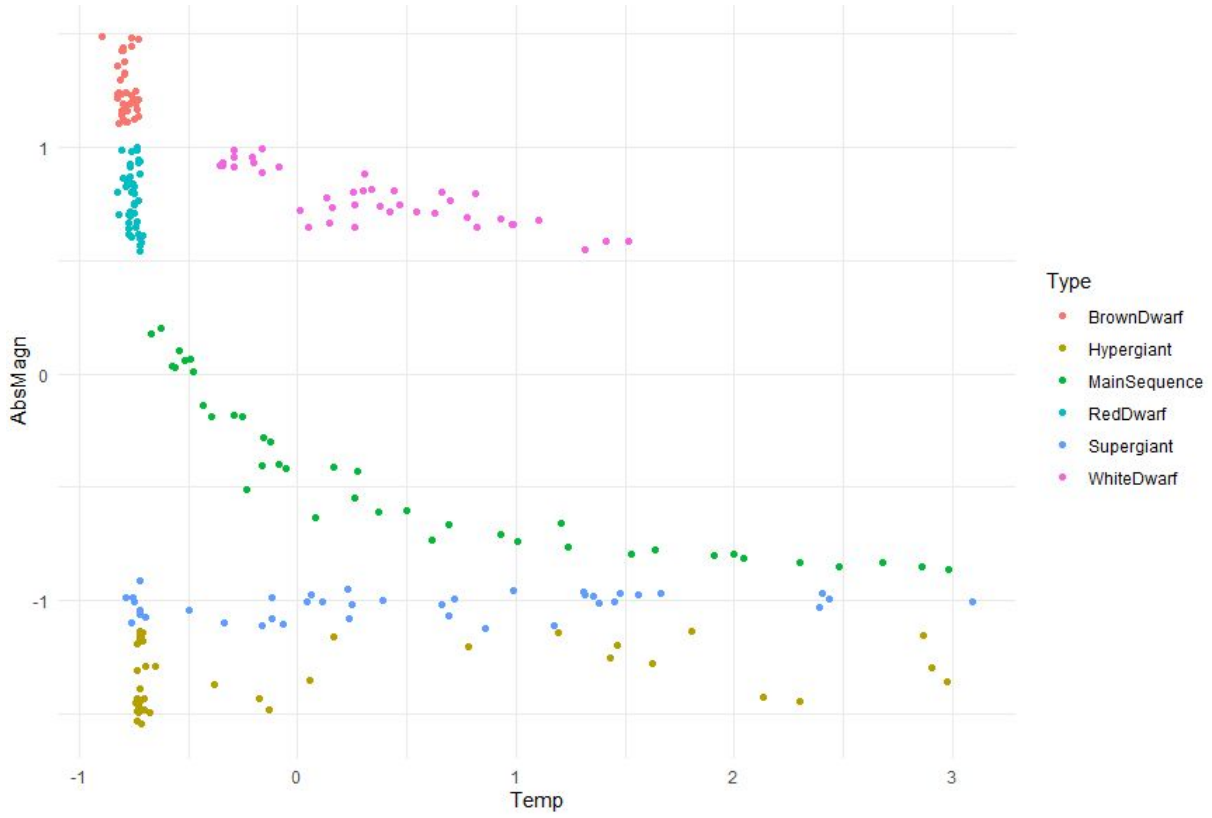

***Figura 4**: features plot - AbsMagn/Temp*

A conferma di ciò, è possibile osservare le distribuzioni del target rispetto alle features AbsMagn e Temp. In *Figura 5* è osservabile come AbsMagn distingue discretamente bene 4 classi su 6, dato il forte overlapping tra WhiteDwarf e RedDwarf. In *Figura 6*, invece, considerando solamente questi due target per facilitare la visualizzazione, si nota che queste due classi sono ben distinte rispetto a Temp.
Questa coppia potrebbe essere quindi essere considerata come punto di partenza per un algoritmo che effettua una classificazione tramite costruzione di iperpiani.

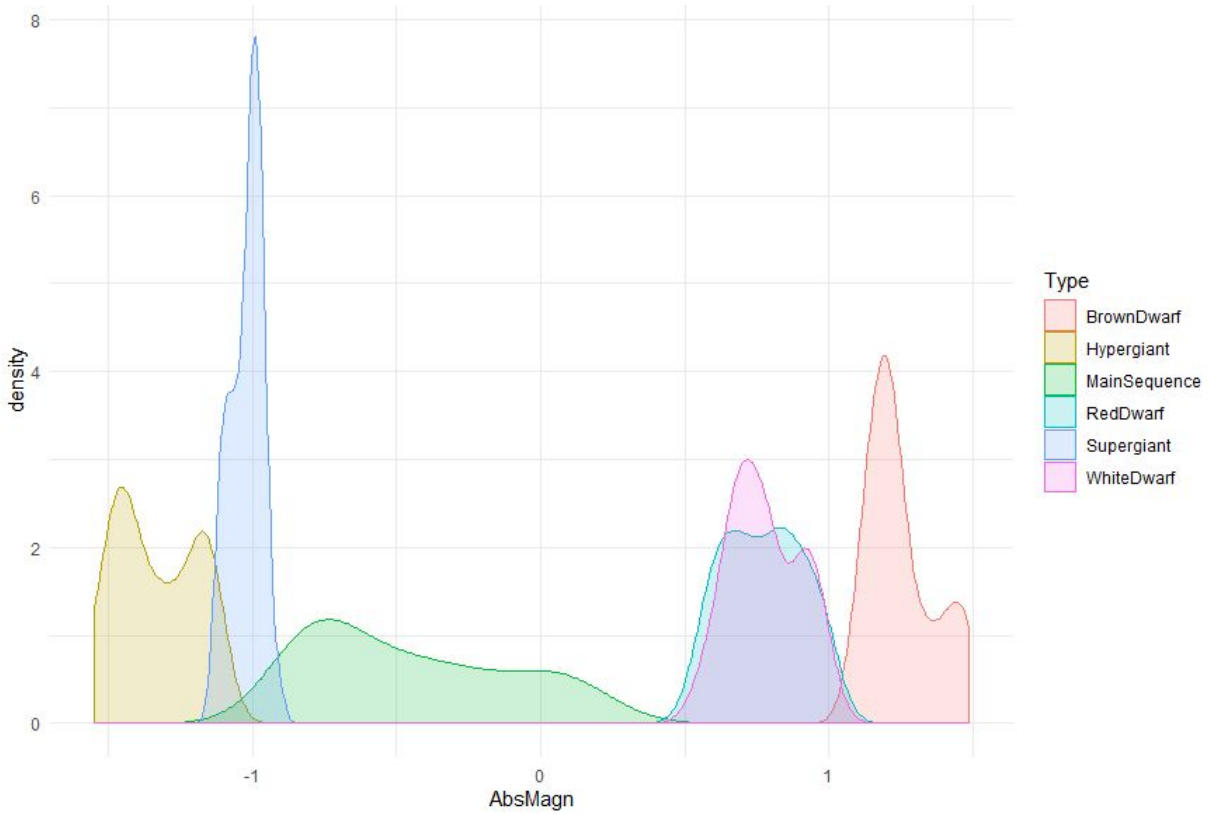

***Figura 5**: distribuzione target - AbsMagn*

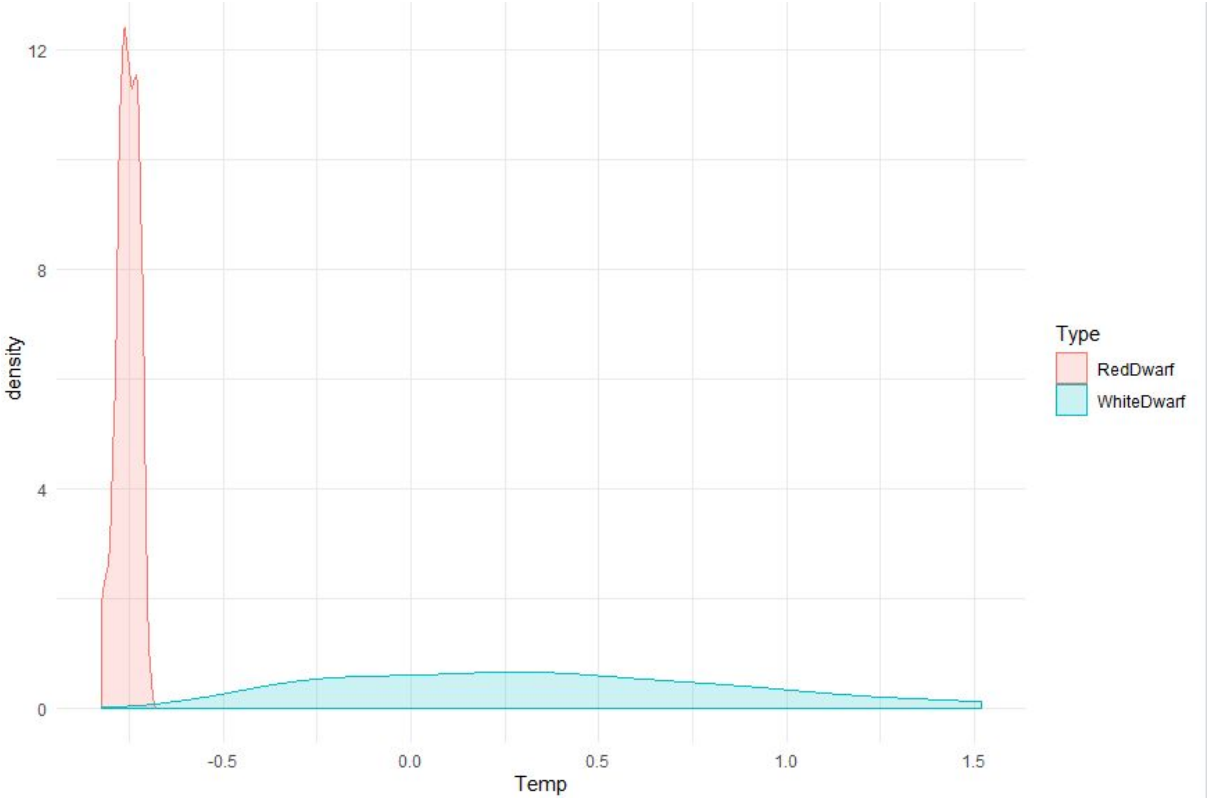

***Figura 6**: distribuzione target - Temp*

Passando invece ad un'analisi che coinvolge anche le variabili categoriche è possibile fare ulteriori considerazioni.
Innanzitutto dal BarPlot in *Figura 7* si può vedere che anche la feature SpectrClass è in grado di distinguere RedDwarf da WhiteDwarf così come osservato in precedenza per Temp, quindi potrebbe essere utile in combinazione con AbsMagn.

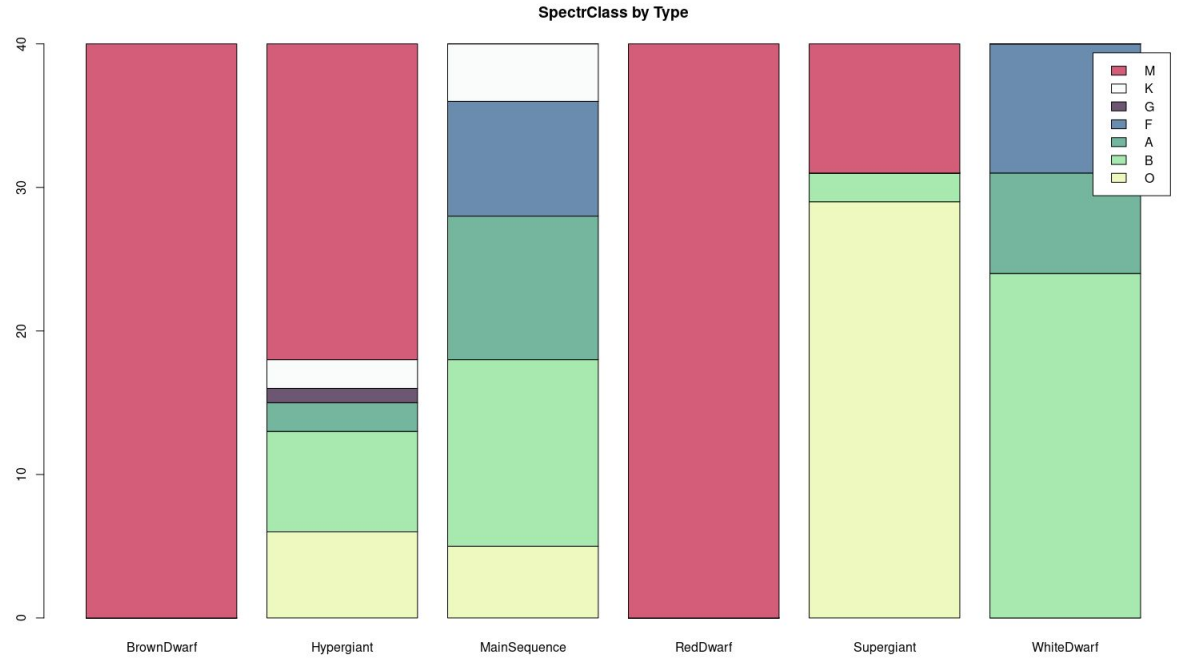

***Figura 7**: distribuzione target - SpectrClass*

A conferma di quanto appena detto, dopo un ulteriore approfondimento sul dominio di interesse, si è potuto riscontrare che AbsMagn e SpectrClass rappresentano un buon classificatore per le stelle [3]; si veda la *Figura 8*. Per verificare che effettivamente tale classificazione vale anche in questo dataset, si è cercato di rappresentare tale grafico. Comparando quindi i due grafici si può notare come in *Figura 9* la distribuzione dei punti rispetti la classificazione mostrata in *Figura 8* .

[3] "Stellar classification - Wikipedia." https://en.wikipedia.org/wiki/Stellar_classification

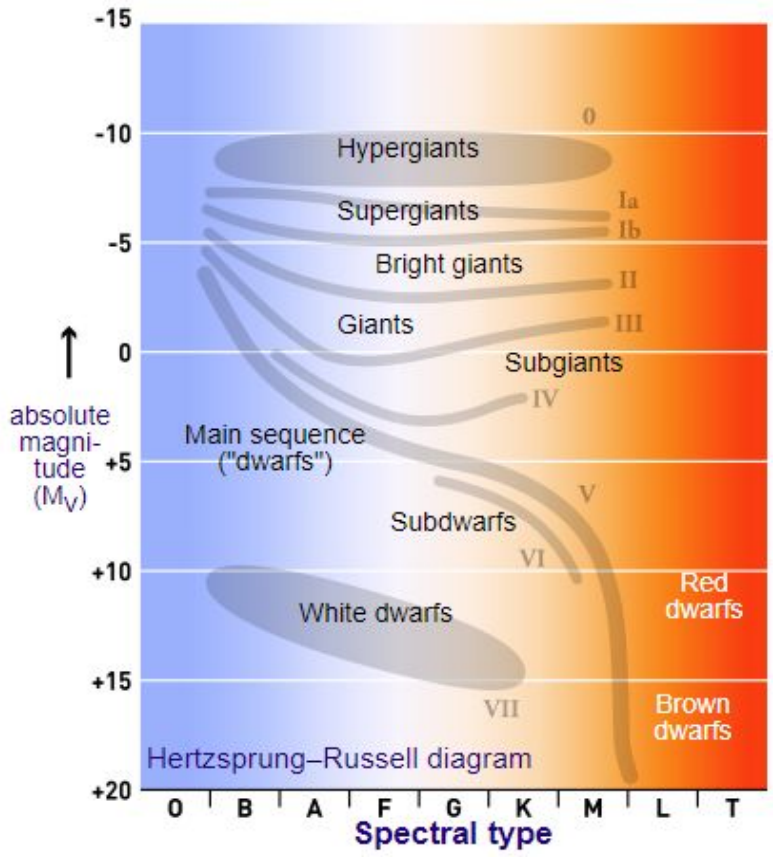

***Figura 8**: stellar classification - Wikipedia*

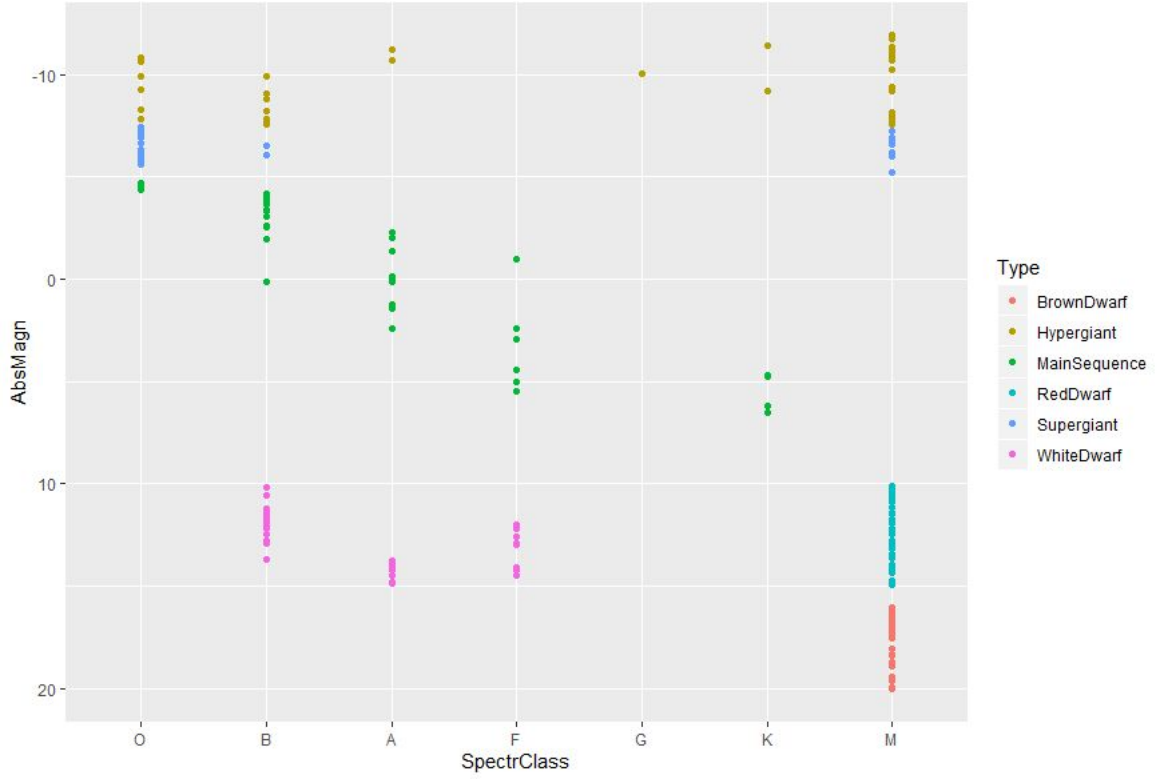

***Figura 9**: stellar classification - Dataset*

Per quanto riguarda la feature Color, invece, possiamo notare la correlazione tra alcuni colori e alcune classi spettrali. Questo si può vedere dalla matrice di correlazione computata dopo la trasformazione degli *n* valori categorici in *n* valori binari tramite one-hot encoding.

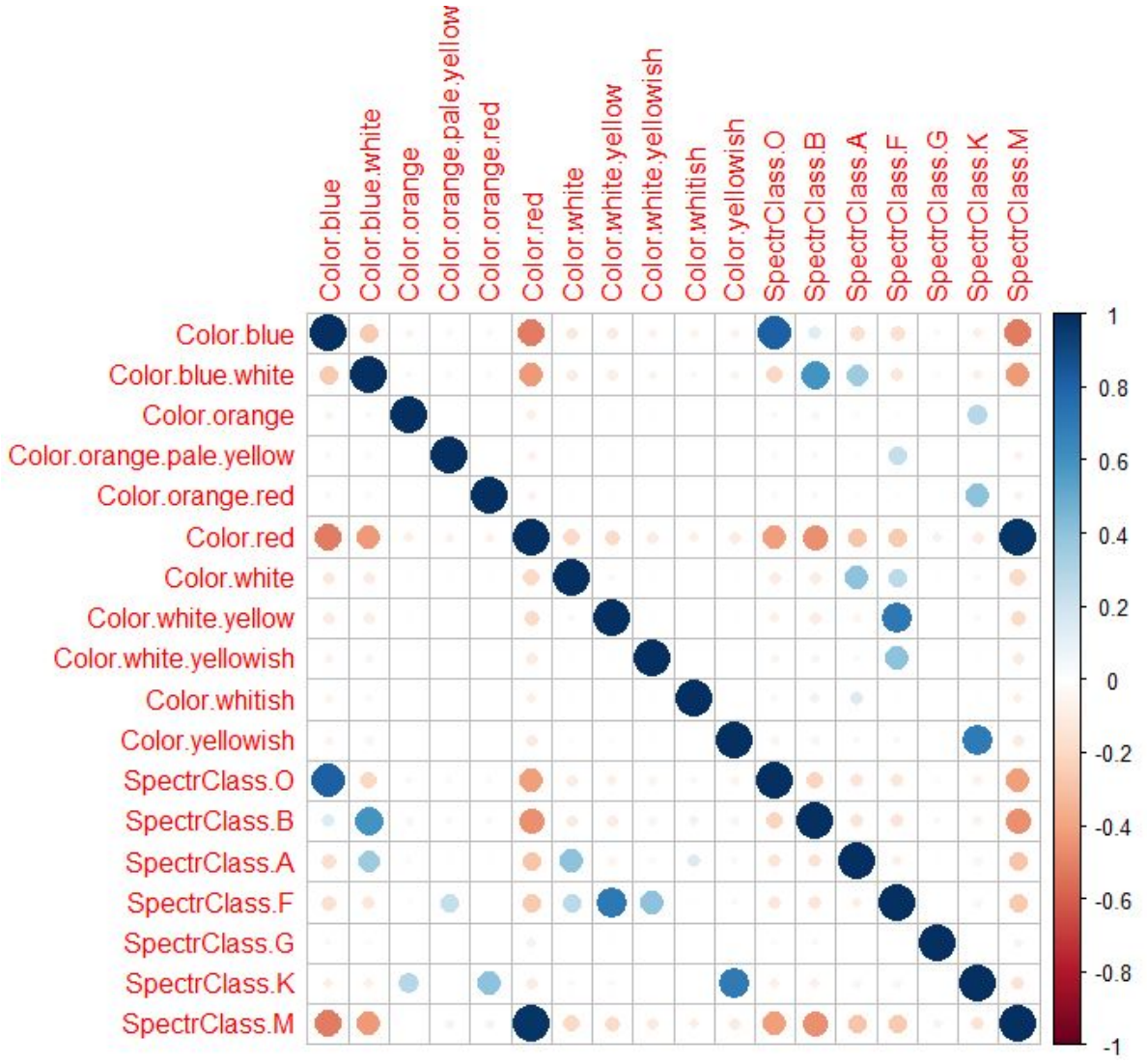

***Figura 10**: matrice di correlazione - Color / SpectrClass*

4 Modelli di machine learning adottati e risultati

Sulla base di quanto osservato durante l'analisi esplorativa, si è deciso di optare per l'utilizzo di SVM, Decision Tree e Random Forest. In particolare l'obiettivo era quello di osservare le differenze tra algoritmi che, operando sullo stesso dataset, effettuassero una classificazione sulla base di features numeriche (SVM) e su features miste (Decision Tree, Random Forest).

Le misure di performance che verranno considerate per il confronto saranno quelle macro, in

quanto il dataset è bilanciato rispetto alle 6 classi che quindi consideriamo di pari importanza.

Il criterio con il quale è stato diviso il dataset è stato quello di mantenere il 70% di istanze per il training set su cui verrà effettuata la 10-fold cross validation, mentre il restante 30% viene riservato al test set su cui valutare le predizioni dei modelli.

4.1 SVM

Le considerazioni fatte nell'analisi esplorativa in merito alle coppie di features che meglio separano le classi target hanno portato alla scelta di un modello SVM per effettuare una classificazione multiclasse tramite costruzione di iperpiani separatori. Come osservato in precedenza, la coppia AbsMagn e Temp sembra rendere possibile una separazione lineare tra le classi utilizzando l'approccio multiclasse one-against-one.
Andando a costruire il modello sul training set tramite una 10-fold cross validation ripetuta per 3 volte, effettuando le predizioni sul test set si può riscontrare un'accuratezza pari circa a 0.93 . Gli errori riguardano in particolar modo la classificazione delle istanze Hypergiant, che vengono diverse volte confuse con Supergiant.
Dato che in *Figura 3* era stato fatto notare che la feature Rad isola molto bene la classe Hypergiant, si è optato per aggiungere al modello SVM anche questa dimensione.
Effettuando nuovamente il training con le stesse modalità, è osservabile un netto miglioramente con un accuracy che sale a 0.98 grazie ad una corretta classificazione delle Hypergiant.
Nel modello finale restituito dal train si osserva che il parametro *cost* viene posto pari a 1.

	Reference					
Prediction	BrownDwarf	Hypergiant	MainSequence	RedDwarf	Supergiant	WhiteDwarf
BrownDwarf	15	0	0	0	0	0
Hypergiant	0	15	0	0	0	0
MainSequence	0	0	10	0	1	0
RedDwarf	0	0	0	9	0	0
Supergiant	0	0	0	0	16	0
WhiteDwarf	0	0	0	0	0	9

***Figura 11**: matrice di confusione - SVM*

	Class: BrownDwarf	Class: Hypergiant	Class: MainSequence	Class: RedDwarf	Class: Supergiant	Class: WhiteDwarf
Precision	1.0	1.0	0.9091	1.00	1.0000	1.00
Recall	1.0	1.0	1.0000	1.00	0.9412	1.00
F1	1.0	1.0	0.9524	1.00	0.9697	1.00
Prevalence	0.2	0.2	0.1333	0.12	0.2267	0.12
Detection Rate	0.2	0.2	0.1333	0.12	0.2133	0.12
Detection Prevalence	0.2	0.2	0.1467	0.12	0.2133	0.12
Balanced Accuracy	1.0	1.0	0.9923	1.00	0.9706	1.00

***Figura 12**: misure di performance per classe - SVM*

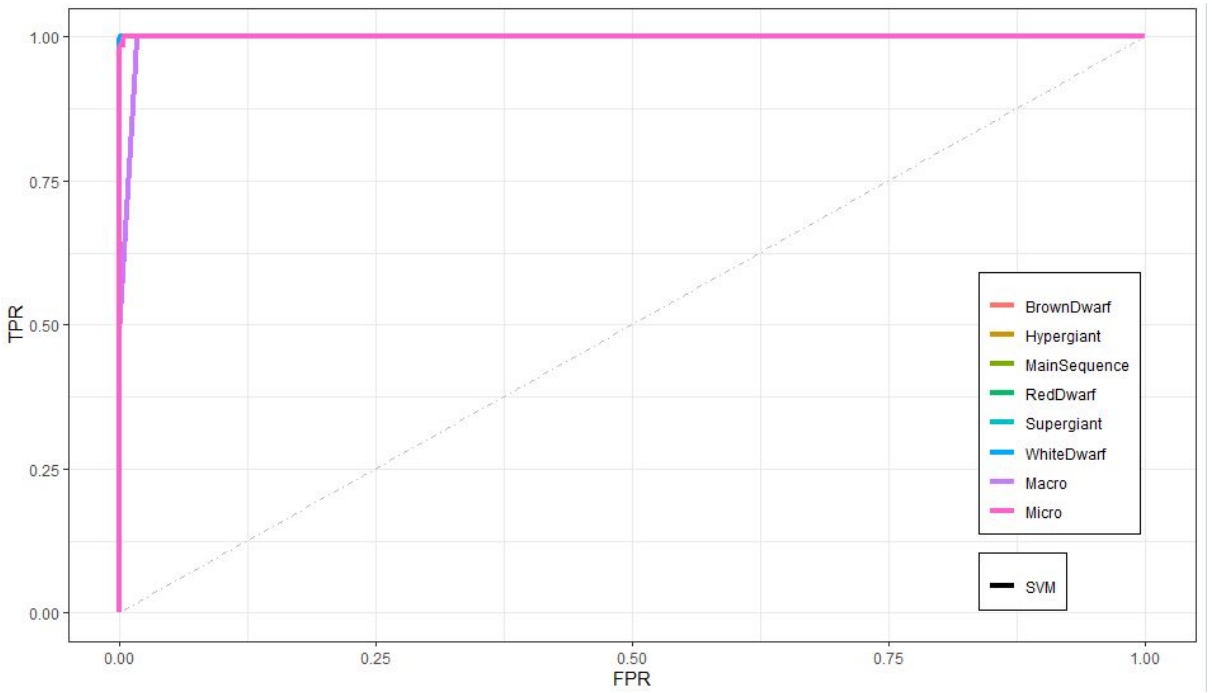

***Figura 13**: curva ROC - SVM*

Di seguito vengono riportate le macro delle misure di performance:

- **precision:** 0.9848485
- **recall:** 0.9901961
- **f-measure:** 0.9870130
- **AUC:** 0.9959891

4.2 Decision Tree

Come secondo modello, data la volontà di utilizzare features categoriche in combinazione con features numeriche, è stato deciso di utilizzare un albero di decisione CART.
Facendo riferimento a quanto appreso nell'analisi delle features categoriche, sono state scelte come features iniziali SpectrClass e AbsMagn.
Andando a costruire il modello sul training set con la medesima modalità di SVM, effettuando le predizioni sul test set si riscontra un'accuratezza di circa 0.77 . Provando ad aggiungere ulteriori features, invece, non si notano miglioramenti. La classe che non riesce ad essere predetta è Supergiant.
Nel modello finale restituito dal train si osserva che il parametro *cp* viene posto pari a 0.172 .

	Reference					
Prediction	BrownDwarf	Hypergiant	MainSequence	RedDwarf	Supergiant	WhiteDwarf
BrownDwarf	15	0	0	0	0	0
Hypergiant	0	15	0	0	17	0
MainSequence	0	0	10	0	0	0
RedDwarf	0	0	0	9	0	0
Supergiant	0	0	0	0	0	0
WhiteDwarf	0	0	0	0	0	9

***Figura 14**: matrice di confusione - Decision Tree*

	Class: BrownDwarf	Class: Hypergiant	Class: MainSequence	Class: RedDwarf	Class: Supergiant	Class: WhiteDwarf
Precision	1.0	0.4688	1.0000	1.00	NA	1.00
Recall	1.0	1.0000	1.0000	1.00	0.0000	1.00
F1	1.0	0.6383	1.0000	1.00	NA	1.00
Prevalence	0.2	0.2000	0.1333	0.12	0.2267	0.12
Detection Rate	0.2	0.2000	0.1333	0.12	0.0000	0.12
Detection Prevalence	0.2	0.4267	0.1333	0.12	0.0000	0.12
Balanced Accuracy	1.0	0.8583	1.0000	1.00	0.5000	1.00

***Figura 15**: misure di performance per classe - Decision Tree*

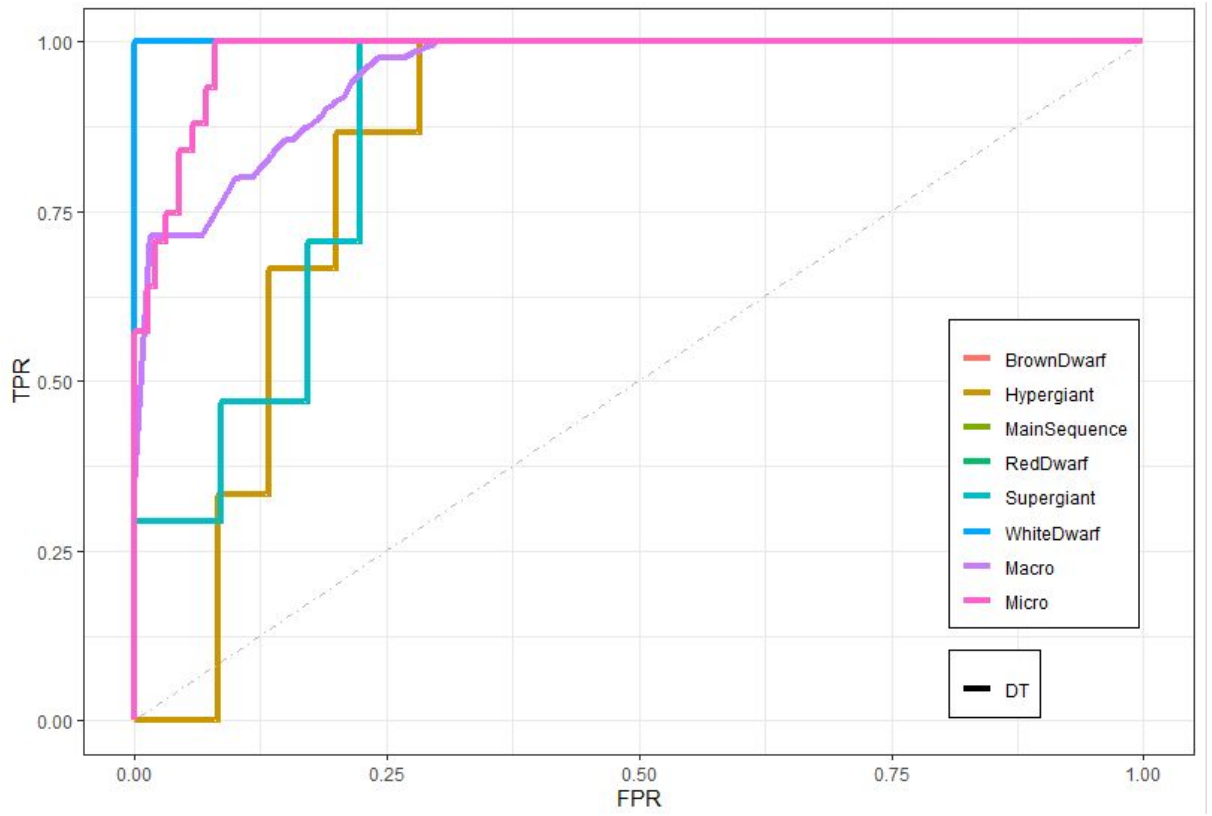

***Figura 16**: curva ROC - Decision Tree*

Di seguito vengono riportate le macro delle misure di performance:

- **precision:** 0.7447917
- **recall:** 0.8333333
- **f-measure:** 0.7730496
- **AUC:** 0.9519066

4.3 Random Forest

Come terzo ed ultimo modello, per pura curiosità in quanto non trattato nel corso, si è deciso di utilizzare Random Forest. L'obiettivo era quello di verificare quanto fosse in grado di migliorare la classificazione rispetto ad un semplice Decision Tree.
Siccome Random Forest è in grado di gestire bene anche dataset con molte più dimensioni costruendo diversi alberi con le varie combinazioni di features, è stato deciso di allenare il

modello con tutte e 6 gli attributi presenti.
Effettuando 3 volte una 10-fold cross validation sul training set, dopo aver effettuato le predizioni sul test set si ottiene un'accuratezza pari a 1.
Nel modello finale restituito dal train si osserva che il parametro *mtry* viene posto pari a 11 .

	Reference					
Prediction	BrownDwarf	Hypergiant	MainSequence	RedDwarf	Supergiant	WhiteDwarf
BrownDwarf	15	0	0	0	0	0
Hypergiant	0	15	0	0	0	0
MainSequence	0	0	10	0	0	0
RedDwarf	0	0	0	9	0	0
Supergiant	0	0	0	0	17	0
WhiteDwarf	0	0	0	0	0	9

***Figura 17**: matrice di confusione - Random Forest*

	Class: BrownDwarf	Class: Hypergiant	Class: MainSequence	Class: RedDwarf	Class: Supergiant	Class: WhiteDwarf
Precision	1.0	1.0	1.0000	1.00	1.0000	1.00
Recall	1.0	1.0	1.0000	1.00	1.0000	1.00
F1	1.0	1.0	1.0000	1.00	1.0000	1.00
Prevalence	0.2	0.2	0.1333	0.12	0.2267	0.12
Detection Rate	0.2	0.2	0.1333	0.12	0.2267	0.12
Detection Prevalence	0.2	0.2	0.1333	0.12	0.2267	0.12
Balanced Accuracy	1.0	1.0	1.0000	1.00	1.0000	1.00

***Figura 18**: misure di performance per classe - Random Forest*

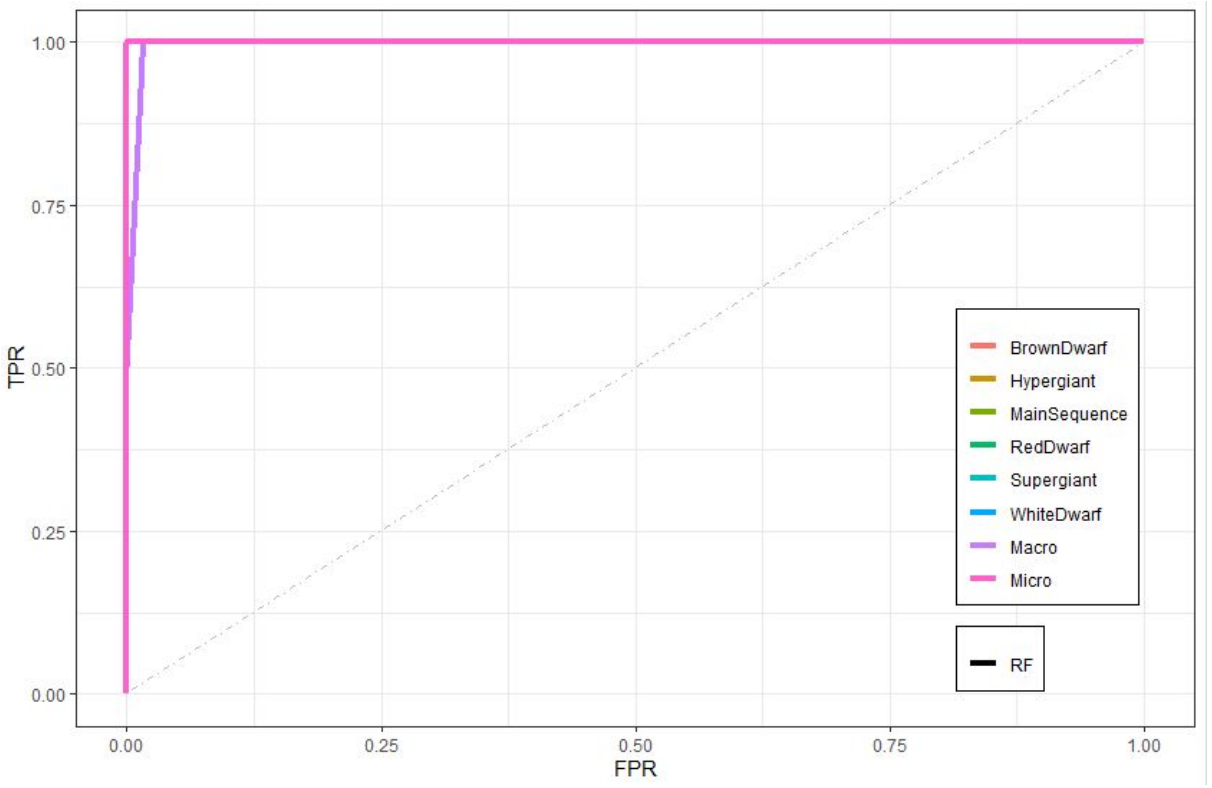

***Figura 19**: curva ROC - Random Forest*

Di seguito vengono riportate le macro delle misure di performance:

- **precision:** 1

- **recall:** 1
- **f-measure:** 1
- **AUC:** 0.9959891

5 Comparazione modelli e conclusioni

Effettuando una comparazione tra le misure ottenute dai tre modelli si ottengono i seguenti risultati.

Model	Precision	Recall	F1	AUC	Acc
SVM	0.9848485	0.9901961	0.9870130	0.9959891	0.9866667
DT	0.7447917	0.8333333	0.7730496	0.9519066	0.7733333
RF	1.0000000	1.0000000	1.0000000	0.9959891	1.0000000

***Tabella 2**: performance a confronto - tabella*

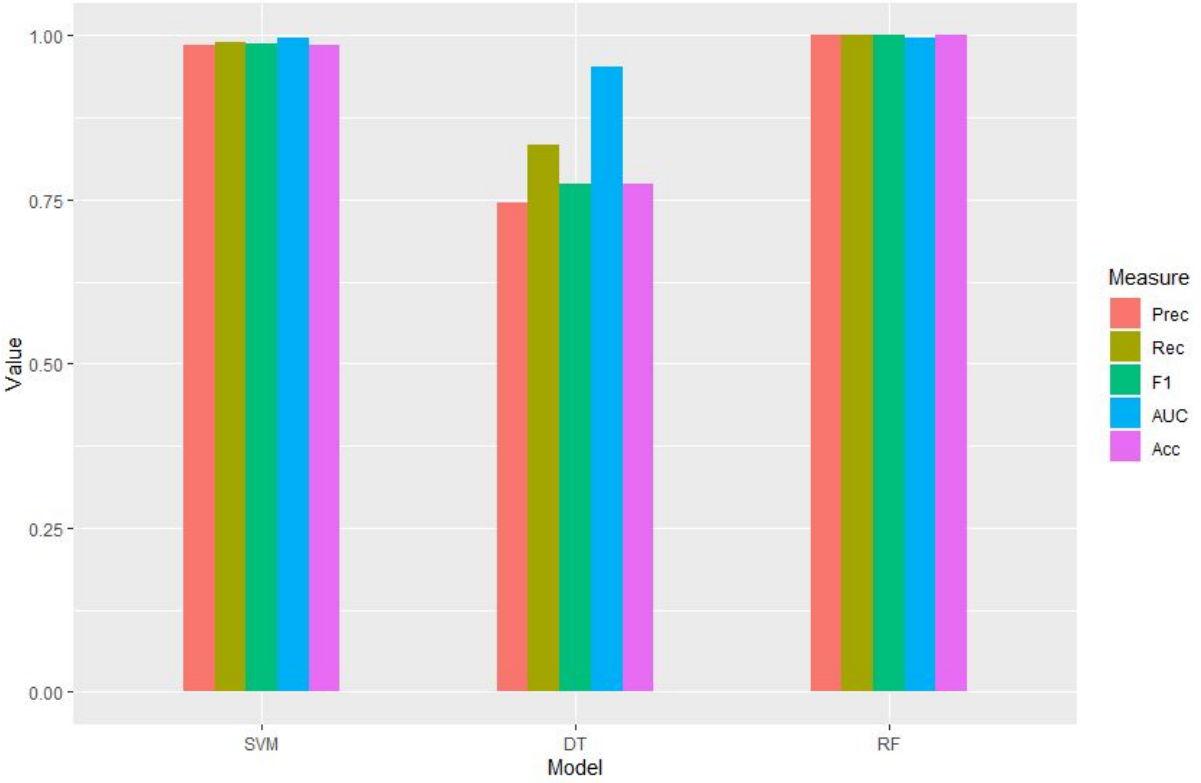

***Figura 20**: performance a confronto - barplot*

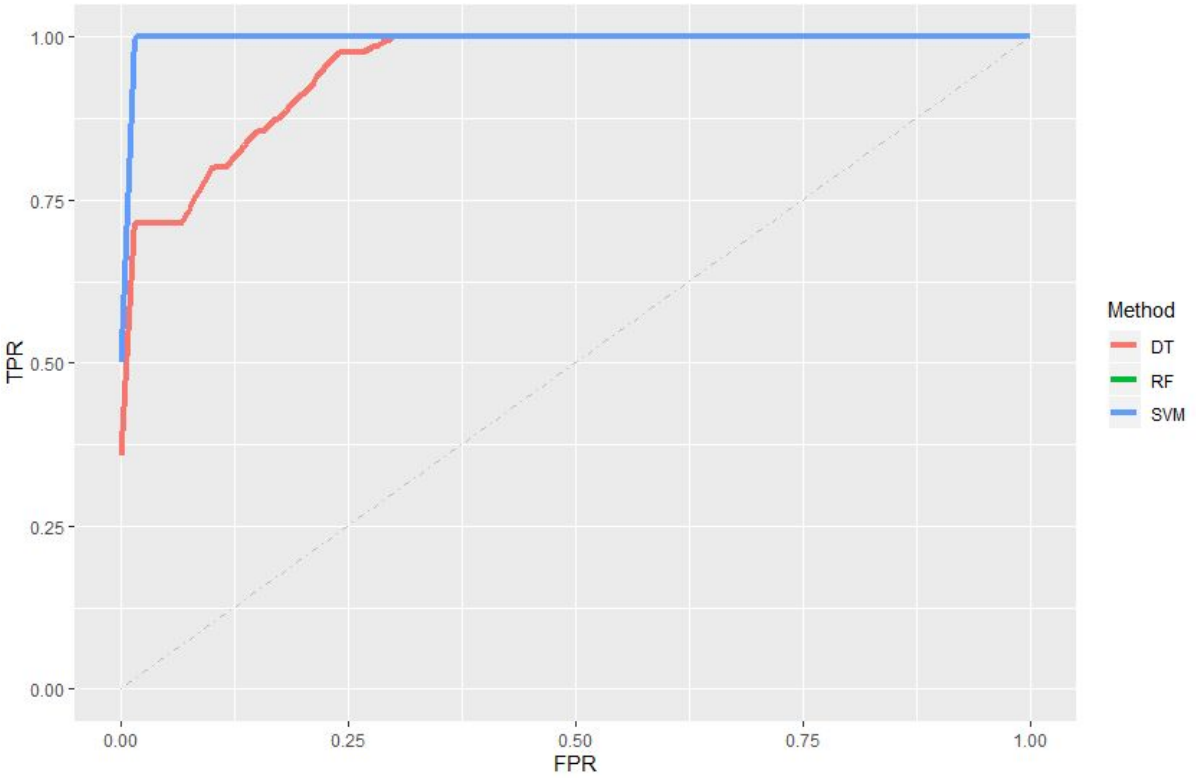

***Figura 21**: macro ROC a confronto*

	Everything	FinalModel
svm	24.25	0.56
dt	15.04	0.02
rf	17.26	0.09

***Tabella 3**: 10-cross validation tempo di training dei modelli*

Osservando i dati ottenuti dalle performance si nota un valore di AUC per la ROC molto vicino a 1 per tutti e tre i modelli.

Per quanto riguarda il modello del Decision Tree, però, si può notare che il valore di f-measure è molto più basso rispetto agli altri (così come per l'accuracy che, data l'equità del dataset, è un buon indicatore per valutare il modello). Ciò è dovuto al fatto che la classe Supergiant non riesce ad essere classificata correttamente, quindi la macro della ROC è fortemente influenzata dalle altre 4 classi.

Per quanto riguarda SVM e Random Forest, si può dire che entrambi i modelli sono molto buoni. In particolare, a parità di AUC, Random Forest raggiunge un valore massimo per le restanti misure di performance.

Anche in merito alle tempistiche Random Forest risulta essere il più rapido ad essere costruito e allenato.