Analisi dei dati 2022/2023: Progetto Finale

David Vencato (Matricola 590954)[1]

Abstract

Partendo da un dataset di 418 pazienti malati di cirrosi biliare al fegato, si vuole prevedere lo stadio istologico della malattia. Per fare ciò sono stati utilizzati due tipi di predittori: le macchine a vettori di supporto (SVM) e le random forest (RF). In seguito, si è valutata l'accuratezza dei risultati con le metriche accuracy, recall e F1-score e loro derivate. Si è quindi fatta un'analisi sull'importanza delle feature tramite il forward stepwise feature e il backward stepwise feature.

Keywords: analisi dati, classificazione, cirrosi epatica

1. Introduzione

Il problema preso in esame rientra nella categoria dei "problemi di classificazione". Infatti, ogni paziente può essere interpretato come un vettore con 20 entrate, tra le quali vi è l'informazione che vogliamo predire cioè lo stadio della malattia.
L'implementazione del codice è stata fatta usando il linguaggio di programmazione Python e il calcolatore su cui sono stati fatti girare i programmi è un MacBook Pro del 2018 (con processore 2,3 GHz Intel Core i5).

2. Preprocessing dei dati

I dati consistono in 418 cartelle cliniche di pazienti affetti da cirrosi epatica, ognuna delle quali contenenti le seguenti informazioni: un "ID" dato dall'ospedale per identificare il paziente; il numero di giorni passati dalla registrazione della malattia fino al giorno in cui c'è stato il trapianto, la morte o la raccolta di questi dati; lo stato del paziente; uso di D-Penicillina (o simili) come parte del trattamento oppure placebo; età; sesso; presenza o meno di ascite, epatomegalia e nevo aracniforme; presenza o meno di edema (e in caso dovuto o no a diuretici); la densità di bilirubina, colesterolo, albumina, rame, fosfatasi alcalina, aspartato transaminasi (SGOT), trigliceridi, piastrine; tempo di protrombina; stadio istologico.

Per prima cosa si sono convertiti i dati qualitativi in dati quantitativi ad eccezione del "ID" identificativo che è stato escluso per l'analisi. Infatti, questo non porta nessun informazione aggiuntiva e anzi rischia di invalidare il processo di predizione. Si è poi notato che alcune cartelle sono deficitarie di qualche dato e dunque sono state compilate facendo sì che rimanesse invariata la mediana dei dati preesistenti. Successivamente, i dati sono stati standardizzati così da poter confrontare tra loro informazioni con unità di misura differenti.
È stato quindi fatto un PCA tridimensionale dei dati riportato in Figura 1.

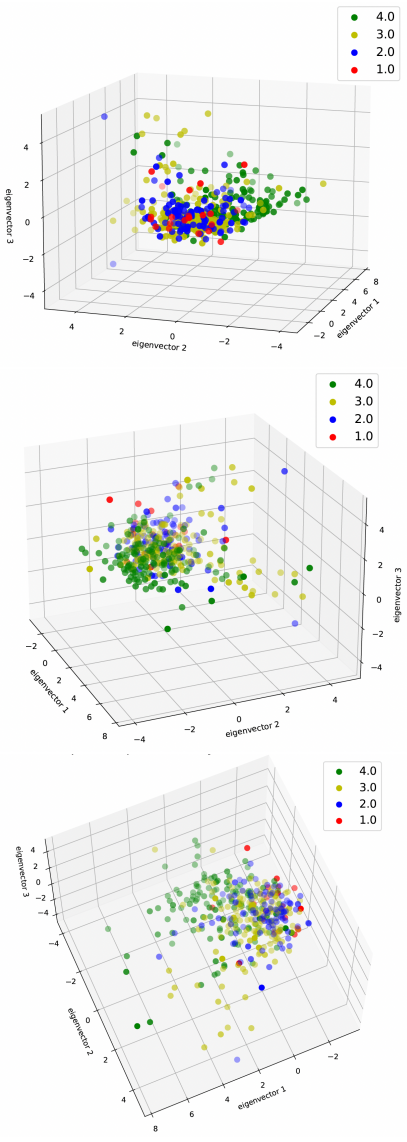

Figure 1: tre diverse angolazione del PCA tridimensionale del dataset

Si nota subito che i punti rappresentanti gli stadi 2 e 3 della malattia sono addensati nella stessa zona di spazio e questo può preannunciare una certa difficoltà nella predizione degli stessi. Per gli stadi 3 e 4 la situazione è leggermente migliore anche se rimangono tante zone di intersezione tra gli spazi occupati.

3. Metriche

La prima metrica usata per l'analisi è stata l' *accuracy*. Questa è definita semplicemente come rapporto tra istanze correttamente classificate e cardinalità del test-set. Questa misura, però, non tiene conto del contesto del problema. Infatti, nel caso "binario" in cui si vuole prevedere se una persona è malata oppure no, si preferisce avere un predittore che, quando sbaglia, dà come risultato che la persona è malata, e non viceversa. Entrano quindi in gioco le definizioni di:

1. *vero positivo* (VP o TP): il predittore dà come malata una persona che effettivamente lo è;
2. *vero negativo* (VN o TN): il predittore dà come sano una persona che effettivamente lo è;
3. *falso positivo* (FP): il predittore dà come malata una persona sana;
4. *falso negativo* (FN): il predittore dà come sana una persona malata.

Questi dati si possono riassumere in una tabella $2x2$ chiamata *confusion matrix* (Figura 2) dove nella diagonale troviamo VP e VN mentre nelle altre due caselle FP e FN.

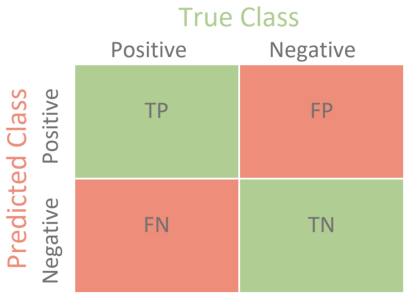

Figure 2: Confusion matrix

Per dare continuità al ragionamento precedente, è dunque utile introdurre la metrica *recall*:

$$recall := \frac{VP}{VP+FN}.$$

È evidente che questa metrica premia maggiormente un numero di falsi negativi basso rispetto all'accuracy. Al contrario la misura *precision*, che ha la stessa formula di recall dove al posto di FN si mette FP, non si adatta alla tipologia di problema; quello che però è possibile fare, per ottenere un'altra misura "equa", è fare la media armonica tra recall e precision ottendendo:

$$F1\text{-}score := \frac{2*recall*precision}{recall+precision}.$$

Si noti che fino ad ora sono state definite metriche che lavorano bene nel caso binario, per questo si deve fare un passo in avanti per poter affrontare il caso "multiclasse". Si suppone dunque di avere n classi; la *confusion matrix* in questo caso sarà una matrice di taglia n dove in coordinata i, j si trova il numero di elementi realmente appartenenti alla classe j che sono stati predetti essere nella classe i. Ora, si definisce la recall della classe i come:

$$recall_i := \frac{VP_i}{VP_i+FN_i}.$$

dove VP_i sono i veri positivi della classe i (l'elemento diagonale di posto i della confusion matrix) e FN_i sono i falsi negativi della classe i (la somma degli elementi della colonna i escluso VP_i). Si procede allo stesso modo per definire l'accuracy e l' $F1-$score della classe i.

A questo punto, in maniera naturale si pone la *recall macro average (ReMaAv)* come la media aritmetica delle recall di ogni classe. Sempre per analogia, si definiscono le "macro average" nel caso "multiclasse" per l'accuracy (che altro non è che l'accuracy totale) e l' $F1-$score. È evidente, però, che in questo modo si trascura la dimensione delle classi stesse, per questo si considera nell'analisi anche la *weighted macro average* cioè la media ponderata con i pesi dati dalla numerosià delle varie classi. In particolare, nel caso della recall, il caso ponderato e l'accuracy coincidono dato che la numerosità di una classe è proprio data dalla somma di veri positivi e falsi positivi ad essa riferiti.

Un altro modo per dare rilevanza alla cardinalità delle classi sarebbe quello di usare la *micro average* di cui però non se ne riporta la costruzione generale poichè nei casi specifici di accuracy, recall e precision dà sempre lo stesso risultato.

4. Predittori

Per cercare di risolvere il problema di classificazione si sono usati i seguenti predittori: le macchine a vettori di supporto (SVM) e le random forest (RF). Il dataset è stato diviso al 20% tra *training set* e *test set*. I risultati con le varie metriche si possono trovare nell'appendice sotto forma di tabelle e nel corso del paragrafo verranno aggiunti dati intermedi per specifici ragionamenti.

4.1. Macchine a vettori di supporto (SVM)

Preliminarmente è stata fatta una "grid search" per cercare gli iperparametri migliori. Per cercare di avere tempi computazionali contenuti la ricerca è stata fatta variando i parametri tra i seguenti valori:

- C: 0.1, 1, 10, 100, 1000;
- *gamma*: 1, 0.1, 0.01, 0.001, 0.0001;
- *kernel*: lineare, polinomiale, radiale e "sigmoid".

Il risultato della ricerca è stato: C=*10*, *gamma*=0.001 e *kernel*=sigmoid.

0	0	0	0
0	0	0	0
6	15	24	16
0	3	4	15

0	0	0	0
1	1	0	0
13	63	112	51
1	10	15	62

Table 1: Confusion matrix per SVM sul test set (sx) e sul training set (dx)

0	0	0	0
0	4	4	1
6	12	19	10
0	2	5	20

1	0	0	0
2	41	0	2
11	24	121	9
1	9	6	102

Table 2: Confusion matrix per RF sul test set (sx) e sul training set (dx)

Scegliendo, dunque, questi parametri per il predittore, si sono ottenute le confusion matrix (per il training set e per il test set) riportate in "Table 1".

Guardando i risultati di questo predittore vengono confermate le congetture fatte con la PCA. Infatti, come si può vedere dalle confusion matrix, lo stadio 1 non è stato mai riconosciuto né nel training set e né nel test set, e solo una volta è stato riconosciuto lo stadio 2 nel training set. Questo è dovuto anche alla bassa cardinalità delle due classi, specialmente lo stadio 1 ha un totale di soli 21 rappresentanti. Si nota anche come effettivamente la classe 3 venga spesso confusa con la classe 2 e la classe 4 con la classe 3.

Guardando le metriche, notiamo che la più alta è l'accuracy, mentre scendono le altre. Ora, l'inefficienza di questi predittori è influenzata molto dalle prime due classi ($Recall_1 = 0\%$ e $Recall_2 = 0\%$ nel test set): la recall della classe 4 nel test set, infatti, è al $48,38\%$ e addirittura è all' $85,71\%$ quella della classe 3. Per quanto riguarda la metrica $F1 - score$ si evidenzia un calo ancora maggiore rispetto alla recall dato che si abbassa in modo significativo l' $F1 - score$ della classe 3 nel test set (53.93%). È comunque rilevante il miglioramento della macromedia pesata per l' $F1 - score$ che restituisce un risultato più veritiero rispetto alla media aritmetica.

4.2. Random Forest (RF)

Anche con questo predittore si è usata una "grid search". Gli iperparametri sono variati tra:

- *profondità massima*: 2, 4, 9, 13, 16, 21;
- *sample split minimo*: 2, 4, 9, 13, 16, 21;
- *criterion*: "gini", "entropy" e "lost loss";
- *numero di stimatori*: 20, 50, 90, 140, 200;
- *bootstrap*: "true" o "false".

La ricerca ha dato come iperparametri migliori i seguenti: *profondità massima*=13, *sample split minimo*=21, *criterion*=entropy, *numero di stimatori*=200 e *bootstrap*=true. Come nel caso precedente, le confusion matrix della random forest con questi iperparametri sono riportate in "Table 2".

La prima cosa che vediamo dalle confusion matrix e dalle tabelle è un problema di overfitting: in ogni caso tra il training set e il test set c'è una discrepanza maggiore del 20%. Di solito, l'overfitting è causato dall'elevatà complessità del modello, dunque un tentativo per cercare di limitarlo potrebbe essere, per esempio, abbassare la profondità della random forest. In ogni caso, i risultati anche sul test set sono migliori rispetto alle macchine a vettori di supporto. Nonostante infatti ci sia la solita influenza "negativa" delle prime due classi nel test set ($Recall_1 = 0\%$, $Recall_2 = 22.2\%$, $F1 - score_1 = 0\%$, $F1 - score_2 = 29.6\%$) si hanno percentuali migliori rispetto all'altro predittore per gli stadi 3 e 4 ($Recall_3 = 67.86\%$, $Recall_4 = 64.52\%$, $F1 - score_3 = 50.67\%$, $F1 - score_4 = 68.97\%$). In effetti dalla confusion matrix si nota come questa random forest riesca a distinguere meglio la classe 4 dalle altre classi, che è anche quello che si era notato nell'analisi tramite il PCA.

5. Importanza dei caratteri

Per valutare l'importanza dei caratteri è stata eseguita un'analisi *forward stepwise feature selection* e *backward stepwise feature selection* nel training set usando come metrica l'accuracy. In appendice si trovano le tabelle per tutti e due i predittori.

Nel primo caso quello che viene fatto è partire da un insieme vuoto e aggiungere di volta in volta la *feature* che migliora maggiormente il risultato; nel secondo caso, al contrario, si parte dall'insieme formato da tutte le caratteristiche e ad ogni passo viene tolta la meno significativa.

Osservando la tabella relativa alla random forest si nota che il colesterolo è una feature importante sia per l'analisi in avanti che all'indietro. Si trova infatti al secondo posto nel forward stepwise e al terzultimo posto nel backward stepwise; in quest'ultimo addirittura fa registrare il delta maggiore tra le percentuali cumulative: questa infatti scende dal 67.78% al 58.97%. Il dato è particolarmente interessante leggendolo in ottica dell'articolo del "Journal of International Medical Research" ([3]) nel quale si mette in evidenza la correlazione proprio tra i livelli di colesterolo e la previsione di morte per la cirrosi epatica cronica. Dall'analisi all'indietro risulta addirittura che senza la protrombina nel dataset la previsione sarebbe migliore.

Per quanto riguarda invece la tabella relativa al predittore SVM è da mettere in evidenzia la caratteristica "stato" (primo posto nella forward stepwise e terzultimo posto nella backward stepwise). Anche questa volta il colesterolo sembra essere una feature davvero importante e ancora nell'analisi all'indietro la protrombina occupa la stessa posizione della classifica con la RF (come prima la predizione sarebbe migliore senza lei); è da notare però che quest'ultima viene messa al terzo posto nell'analisi in avanti.

Evidentemente, la semplicità dei due modelli è "pagata" con una forte instabilità e, tranne che al primo passo, non analizzando tutti i possibili sottoinsiemi, non è detto restituiscano la

combinazione migliore.

6. Conclusioni

Le metriche pesate sembrano essere migliori rispetto a quelle aritmetiche dato un dataset con classi così poco numerose e distrbuite non equamente. L'analisi è infatti molto influenzata dagli scarsi risultati sullo stadio 1 e in parte sulla classe 2. È però interessante come i predittori sulle classi più popolose (che sono quelle più distinguibili per il PCA) si siano comportati meglio proprio sulla misura recall che è anche quella più pertinente al contesto.

Anche l'analisi sulle feature si è rilevata essere piuttosto utile per mettere in risalto alcune caratteristiche che con molta probabilità sono davvero importanti per la predizione (ad esempio il colesterolo) mentre altre sono più marginali (ad esempio la protrombina). Per contro ci sono altre caratteristiche che restano in una zona di chiaroscuro dove è difficile fare considerazioni con strumenti semplici e dal costo computazionale non elevato.

7. Link

Di seguito il link alla pagina di Google Colab: clicca qui.

8. Bibliografia e sitografia

[1] Understanding Machine Learning - From Theory to Algorithms. S. Shalev-Shwartz, and S. Ben-David. Cambridge University Press;

[2] Metrics for Multi-Class Classification: an Overview. Margherita Grandini, Enrico Bagli, Giorgio Visani. https://arxiv.org/abs/2008.05756;

[3] HDL-C levels added to the MELD score improves 30-day mortality prediction in Asian patients with cirrhosis. Yue Wang1,*, Wenjuan Shen2,*, Fang Huang3,*, Chenyan Yu2,*, Liting Xi2, Jingwen Gao2, Minyue Yin2, Xiaolin Liu2, Jiaxi Lin2, Lu Liu2, Huixian Zhang2, Jinzhou Zhu and Yu Hong2. https://journals.sagepub.com/doi/epub/10.1177/03000605221109385.

[4] https://www.kaggle.com/datasets/fedesoriano/cirrhosis-prediction-dataset.

Accuracy		
Predittore	*Training set (%)*	*Test set (%)*
SVM	53.19	46.99
RF	79.03	53.09

Recall				
Predittori	*Recall Macro Average*		*Recall Weighted Macro Average*	
	Training set (%)	*Test set (%)*	*Training set (%)*	*Test set (%)*
SVM	33.67	33.53	53.19	46.99
RF	61.01	38.65	79.03	53.09

F1-score				
Predittori	*F1-score Macro Average*		*F1-score Weighted Macro Average*	
	Training set (%)	*Test set (%)*	*Training set (%)*	*Test set (%)*
SVM	31.38	27.63	45.41	39.34
RF	62.38	37.31	77.06	50.49

Predittore: RF, Metrica: Accuracy			
Forward-stepwise		Backward-stepwise	
Carattere	*% cumulativa*	*carattere*	*% cumulativa*
N° giorni	62.00	Protom.	81.16
Colest.	67.78	Stato	82.07
SGOT	71.12	Pennic.	82.37
Biliru.	72.95	N° giorni	81.46
Sesso	74.47	Trigl.	81.46
Trigl.	75.99	Rame	80.55
Fosfat.	76.29	Ascite	78.42
Rame	76.90	Edema	79.03
Protom.	78.72	Sesso	79.94
Stato	79.64	Nevo	78.16
Età	80.24	SGOT	78.16
Edema	79.03	Biliru.	76.60
Nevo	79.33	Fosfat.	73.56
Ascite	79.33	Età	72.34
Albumina	78.42	Epatome.	70.21
Pennic.	80.34	Colest.	67.78
Epatome.	79.33	Albumina	58.97
Piastri.	78.12	Piastri.	58.97

Predittore: SVM, Metrica: Accuracy			
Forward-stepwise		Backward-stepwise	
Carattere	*% cumulativa*	*carattere*	*% cumulativa*
Stato	47.42	Protom.	82.07
Età	49.54	Pennic.	82.37
Protom.	52.58	Ascite	81.46
Nevo	53.19	Fosfat.	81.46
Trigl.	53.50	Edema	80.55
Pennic.	53.50	Biliru.	78.42
Colest.	53.50	Sesso	81.16
N° giorni	54.40	Nevo	79.03
Epatome.	55.02	SGOT	79.94
Sesso	55.02	Trigl.	78.16
Ascite	55.02	Età	78.16
Edema	55.02	Epatome.	76.60
Fosfat.	55.62	Rame	73.56
Biliru.	55.02	N° giorni	72.34
SGOT	55.02	Piastri.	70.21
Piastri.	55.02	Stato	67.78
Rame	55.02	Colest.	58.97
Albumina	53.19	Albumina	58.97