Relazione Machine Learning
Breast Cancer Wisconsin DataSet

Marco Bolpagni, Tommaso Ulivieri
marco.bolpagni@gmail.com - tommaso.ulivieri@gmail.com

Introduzione

L'obiettivo del progetto è di risolvere un problema di classificazione binaria relativo al dataset 'Breast Cancer Wisconsin DataSet': un dataset diagnostico per la classificazione del tumore al seno. All'interno del dataset ogni paziente ha un identificativo (id) ed è descritto da 30 caratteristiche numeriche e dalla classe (diagnosis) che rappresenta l'esito della diagnosi.

Il dataset oggetto di studio è costituito da 455 casi.
Di seguito si riporta l'elenco delle *features* presenti nel dataset relative al nucleo cellulare analizzato:

- radius - media delle distanze dal centro ai punti sul perimetro;
- texture - deviazione standard dei valori della scala dei grigi;
- perimeter - perimetro;
- area - area;
- smoothness - variazione locale;
- compactness - perimeter^2/area - 1.0;
- concavity - gravità delle porzioni concave del contorno;
- concave points - numero di porzioni concave del contorno;
- symmetry - simmetria;
- fractal dimension - dimensione frattale.

Per ognuna delle *features* sono state calcolate la media, l'errore standard e il peggiore (o più grande) ottenendo così un totale di 30 caratteristiche per ogni immagine.

Il classificatore implementato deve prevedere, sulla base della caratteristiche di input, se il tumore analizzato è benigno o maligno.

Data Understanding

L'analisi del dataset consente di osservare un lieve sbilanciamento delle diagnosi con una prevalenza di casi di tumori benigni (62.64%) rispetto ai maligni (37.36%). Si osserva inoltre la presenza di valori mancanti con una distribuzione randomica. Dall'analisi esplorativa del dataset si è osservata la presenza di due *features* con valori molto più grandi rispetto alle altre, cosa che potrebbe distorcere le performance del classificatore.
È inoltre possibile rilevare la presenza di alcune *features* altamente correlate.
Infine l'analisi delle distribuzioni ha evidenziato una differenza tra casi di tumori benigni e maligni e la presenza di outlier in alcune features.

Metodologia e Data preparation

Al fine di valutare la performance del classificatore creato è stato deciso di conservare una parte del dataset ("test set") che non sarà utilizzato in fase di addestramento degli algoritmi. **Lo split del dataset** è stato effettuato con una proporzione di 80-20 (training set-test set) ed è stato stratificato in modo da mantenere la proporzione dell'outcome della diagnosi.

Sono stati svolti alcuni tentativi preliminari di ***features selection*** che hanno prodotto modelli più semplici con performance lievemente inferiori. Tuttavia considerato il contesto applicativo del classificatore è stato scelto di dare priorità all'accuracy piuttosto che alla semplicità del modello.
Considerata la presenza di *features* altamente correlate, è stato deciso di implementare una tecnica di **riduzione della dimensionalità**, nello specifico l'Analisi delle componenti principali che consente di ottenere un minor numero di dimensioni incorrelate. È stato dimostrato in letteratura che la PCA consente di ridurre il "rumore" e ottimizzare i tempi di computazione (Gokgoz & Subasi, 2014)[1].

Vista la presenza ridotta di **dati mancanti** e la loro distribuzione (MAR) è stato deciso di procedere con l'imputazione degli stessi, testando diverse modalità in fase di ottimizzazione del modello: media, mediana e vicino più prossimo (KNN). Il fit dell'imputer è stato effettuato sul training set per evitare il fenomeno del data leakage[2].

Dall'analisi esplorativa del dataset si è osservata la presenza di due *features* con valori molto più grandi rispetto alle altre, cosa che potrebbe distorcere le performance del classificatore.
Per questo motivo è stato deciso di effettuare **lo scaling dei dati** utilizzando diverse tecniche in fase di ricerca dell'algoritmo più performante: minmax, standard scaler, robust scaler.
Come per la fase di imputazione il fit dello scaler è stato implementato sul training set per evitare il fenomeno del data leakage.

Nella fase di costruzione e verifica delle performance del modello, è stato deciso di implementare una **Pipeline combinata con la Grid Search**. Questo approccio ha consentito di testare in un solo passaggio diversi possibili classificatori analizzando le performance al variare dei singoli iperparametri valutati.
Per ridurre il carico computazionale è stata testata l'efficacia della ricerca degli iperparametri attraverso un approccio bayesiano volto all'ottimizzazione. L'approccio bayesiano, a differenza della random search o della grid search, tiene traccia dei risultati delle valutazioni passate e li usa per scegliere gli iperparametri in base alla probabilità di un punteggio sulla funzione obiettivo:

$$P(score \mid hyperparameters)$$

[1] Gokgoz, E., & Subasi, A. (2014). Effect of multiscale PCA de-noising on EMG signal classification for diagnosis of neuromuscular disorders. Journal of Medical Systems, 38(4).
[2] Il fenomeno del "data leakage" si verifica quando il modello viene addestrato su dati che non dovrebbero essere inclusi nel training set o che comunque non sarebbero disponibili in simulazioni con dati reali. (https://www.educative.io/edpresso/data-leakage-in-machine-learning)

Ad alto livello, i metodi di ottimizzazione bayesiani sono efficienti perché scelgono i successivi iperparametri in modo informato.
Questo modello è stato implementato con la libreria *scikit-optimize 0.8.1*. Nonostante i risultati promettenti, è stato deciso di abbandonare questa strada in quanto la libreria non permette l'ottimizzazione dei Multilayer Perceptron a causa di una mancanza nella gestione delle liste di tuple (non sarebbe stato possibile modificare il numero di neuroni e strati nascosti).

È stata implementata una **cross validation** con 5 fold[3] per ottenere una maggiore generalizzabilità dei risultati e avere un buon trade off tra bias e varianza evitando i fenomeni di overfitting e underfitting[4].

Sulla base della letteratura esistente relativa alla classificazione binaria di dati di natura medica (Mittal & Gill, 2018)[5] è stato deciso di testare i seguenti **classificatori**:

- Logistic Regression
- Gaussian Naïve Bayes
- K-Nearest Neighbour
- Support Vector Machine
- Random Forest
- Multilayer Perceptron

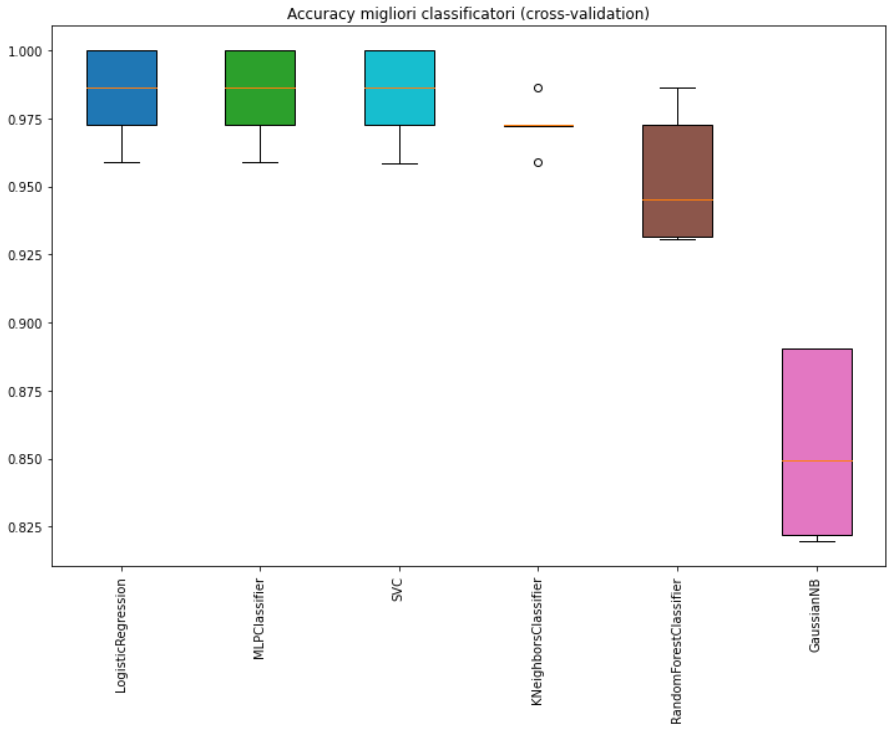
Data la natura lievemente sbilanciata del dataset, ove possibile sono stati utilizzati algoritmi di apprendimento penalizzati che aumentano il costo degli errori di classificazione sulla classe meno rappresentata (cost-sensitive training).

Per l'implementazione del progetto è stata utilizzata la libreria *scikit-learn 0.23.2* con *Python 3.7.6.*

[3]I 5 fold rappresentano un buon trade off tra costo computazionale e qualità del risultato.
[4]Müller, A. C. & Guido, S. (2016). Introduction to machine learning with Python: A Guide for Data Scientists. Bejing: Oreilly et Associates.
[5]Mittal, P., & Gill, N.. (2014). A Comparative Analysis Of Classification Techniques On Medical Data Sets. International Journal of Research in Engineering and Technology, 03(06), 454-460.

Valutazione del modello e conclusioni

L'analisi svolta ha consentito di individuare 3 classificatori con un livello di accuracy in cross validation pressoché equivalente. I classificatori in questione sono Logistic Regression, Multilayer Perceptron e Support Vector Machine.



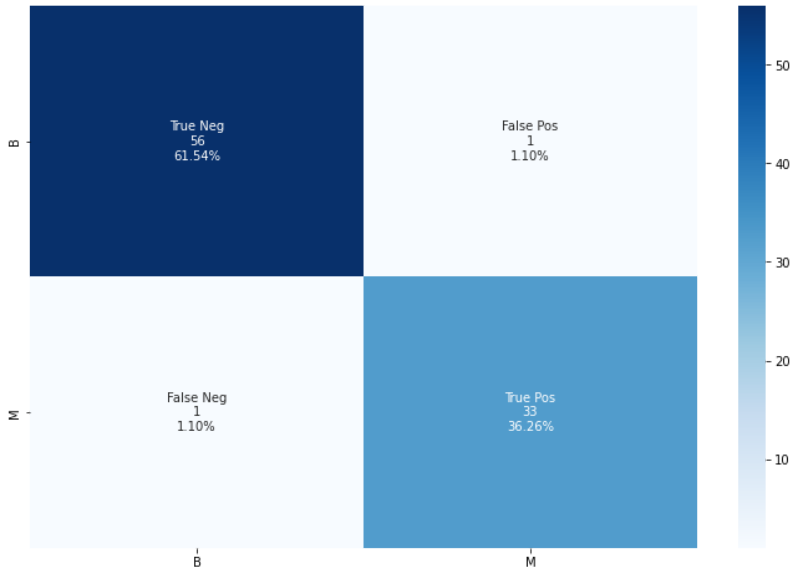
A parità di performance, è stato scelto come modello migliore per il problema in esame il modello più semplice (principio del rasoio di Occam) tra i tre più performanti, ovvero:

- Strategia di imputazione dei dati mancanti: valore medio
- Metodo di scaling: standard scaler (z-score)
- Algoritmo di classificazione: Logistic Regression con solver liblinear, regolarizzazione Lasso, C=1 e bilanciamento delle classi

I risultati ottenuti confermano quanto emerso nello studio "*A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models"* (Christodoulou et al., 2019)[6], ovvero che per problemi di classificazione binaria di tipo medico relativamente semplici la regressione logistica, opportunamente ottimizzata, può avere performance in linea con quelle di algoritmi di machine learning più complessi.

[6] Christodoulou, E., Ma, J., Collins, G. S., Steyerberg, E. W., Verbakel, J. Y., & Calster, B. V. (2019). A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. Journal of Clinical Epidemiology, 110, 12-22.

La seguente matrice di confusione mostra le performance del modello sul test set e presenta un'equidistribuzione degli errori di classificazione (falsi positivi-falsi negativi).



In conclusione, considerato il contesto di ricerca, potrebbe essere interessante utilizzare come parametro di valutazione della performance e per il tuning degli iperparametri la *recall*[7] che, a differenza dell'*accuracy*, pone l'accento sull'importanza di ridurre i falsi negativi. Trattandosi di una patologia potenzialmente mortale riteniamo molto rischioso produrre un classificatore che tolleri i falsi negativi.

[7] Recall = True Positive / (True positive+False Negative)