Advanced Machine Learning: Assignment #1

Fabrizio D'Intinosante

Università degli Studi di Milano Bicocca — October 19, 2019

Introduzione

Il dataset contiene informazioni riguardo le inadempienze di pagamento, variabili demografiche, dati di credito, storico dei pagamenti ed estratti conto di possessori di carte di credito di Taiwan da Aprile 2005 a Settembre dello stesso anno, distribuite su **24 colonne** e **27000 righe**. L'obiettivo è quello di predire le inadempienze di pagamento attraverso l'utilizzo di una *neural network*.

1 Esplorazione e preprocessing

Come primo passaggio si è provveduto ad esplorare il *dataset*, le distribuzioni delle variabili esplicative e quella della variabile obiettivo. Come si è potuto osservare tramite visualizzazione la variabile obiettivo presenta un forte sbilanciamento.

Per questa ragione si è deciso di procedere alla realizzazione di tre modelli NN:

- un modello semplice, che tiene conto dello sbilanciamento solo attraverso l'aggiustamento dei pesi;
- un modello addestrato su dati sottoposti ad *oversampling* (SMOTE);
- un modello addestrato su dati sottoposti ad *undersampling*.

Le *features* disponibili sono state innanzitutto suddivise in *features* numeriche e categoriche per poter distinguere la strategia di *preprocessing* a loro applicata: mentre infatti per le prime si è deciso di procedere con una semplice standardizzazione (processo che porta la media $\mu = 0$ e la varianza $\sigma^2 = 1$ delle *features*), per le variabili qualitative, in questo caso tutte nominali, si è preferito procedere con la strategia *One-hot encoding* (OHE), ovvero ridurre ogni livello delle variabili ad una *dummy*.

NB: Durante la creazione delle *dummy* si è proceduto ad eliminare il primo livello di ognuna, così da evitare di incorrere in problemi di multicollinearità.

Una volta applicate le tecniche di *preprocessing* su entrambi i gruppi di covariate si è proceduto a riunificarle in un unico *numpy array* concatenandole. Successivamente per tutti e tre i modelli i dati sono stati suddivisi in *training set* e *validation set* con una proporzione di 80% e 20% (fatta eccezione per il modello con dati trattati con l'*undersampling* per cui si è utilizzata una proporzione 90% e 10% per evitare che dopo l'*undersampling* il *training set* risultasse inferiore al *validation set*. Il *validation set* oltre ad esser stato utilizzato per l'ottimizzazione degli iperparametri, restando al di fuori del processo di *training* è stato utilizzato anche per la valutazione complessiva del modello attraverso *classification report*.

NB: Per il secondo ed il terzo modello, ovvero quelli trainati con dati sottoposti ad *oversampling* e *undersapling*, queste tecniche sono state applicate successivamente alla suddivisione tra *training set* e *validation set* così da permettere al modello di apprendere a generalizzare al meglio i dati.

2 Modelli

Come detto in precedenza i modelli realizzati sono tre: un modello con *balancing* attraverso i pesi, uno con trattamento dei dati attraverso *oversampling* ed un ultimo con trattamento dei dati tramite *undersampling*.

```
confusion matrix:
 [[3987  218]
 [ 748  447]]

precision:
 [0.84202746 0.67218045]

recall:
 [0.94815696 0.37405858]

f-score:
 [0.89194631 0.48064516]

              precision    recall  f1-score   support

           0       0.84      0.95      0.89      4205
           1       0.67      0.37      0.48      1195

    accuracy                           0.82      5400
   macro avg       0.76      0.66      0.69      5400
weighted avg       0.80      0.82      0.80      5400
```

Figure 1: *classification report* del modello 1

Tutti e tre i modelli condividono tutti gli iperparametri fatta eccezione per il valore di *rate* dei *dropout layer*:

- 100 epoche;
- tre *hidden layers* in ordine da 32, 16 e 16 neuroni. Il numero di *layer* e di neuroni è stato scelto dopo aver effettuato diverse prove ed aver verificato che fosse un buon compromesso per non complicare inutilmente il modello implementando però il suo potere predittivo;
- *activation function hidden layers:* ReLU. Scelta standard, si è rivelata la più adeguata;
- *activation function output layer: Sigmoid*. Trattandosi di una classificazione binaria si è trattato di una scelta obbligata considerando che il vettore y è stato mantenuto in forma dummy e non convertito in forma one-hot per cui sarebbe stato necessario utilizzare una *Softmax*;
- pesi personalizzati per unbalancing (utili solo per il primo modello, per gli altri risultano inutili avendo effettuato balancing a priori sui dati tramite *under/oversampling*. Per bilanciare lo sbilanciamento della variabile y i pesi così definiti assegnano, nella *loss function* un maggiore peso alla classe sottorappresentata così da bilanciare l'effetto.)
- *optimizer:* adam. Anche questa una scelta piuttosto standard, ottiene buone prestazioni essendo un algoritmo di ottimizzazione adattivo;
- *loss function: binary_crossentropy*. Scelta obbligata trattandosi di stimare valori [0,1];
- *regularization:* tre *dropout layers* ed *early stopping* con una *patience* di 20. I rate dei *dropout layers* sono molto piccoli o addirittura azzerati a seconda del modello ma sono stati comunque implementati per poter testare meglio i diversi iperparametri della rete. L'unione con l'*early stopping* è stata fatta per scongiurare il rischio di overfitting del modelli.

2.1 Valutazione dei modelli

Come si vede in fig. 1 il modello bilanciato attraverso i pesi è il migliore per valore di *accuracy* ed *weighted avg f1-score*. Il modello con dati trattati attraverso *oversampling* invece, il cui *classification report* è presentato in fig. 2, presenta prestazioni inferiori anche se la *f1-score measure* della classe di y meno rappresentata risulta migliore rispetto al modello precedente. Il modello con dati trattati invece tramite *undersampling*, presentato in fig. 3 risulta tendenzialmente migliore rispetto a quello con dati trattati con *oversampling*, ma comunque inferiori rispetto al primo modello.

3 Conclusioni

In conclusione, si può affermare che il modello vincitore risulti essere quello presentato in fig. 1, ovvero il modello con dati non corretti rispetto all'*unbalancing* presente nella variabile obiettivo y, ma che tratta questo elemento attraverso la correzione dei pesi da sottomettere alla *loss function* così da considerare in modi distinti le osservazioni classificate come 0 e quelle classificate come 1.

```
confusion matrix:
 [[3364  841]
 [ 479  716]]

precision:
 [0.87535779 0.4598587 ]

recall:
 [0.8        0.59916318]

f-score:
 [0.8359841  0.52034884]

              precision    recall  f1-score   support

           0       0.88      0.80      0.84      4205
           1       0.46      0.60      0.52      1195

    accuracy                           0.76      5400
   macro avg       0.67      0.70      0.68      5400
weighted avg       0.78      0.76      0.77      5400
```

Figure 2: *classification report* del modello 2 (oversampling)

```
confusion matrix:
 [[1690  413]
 [ 224  373]]

precision:
 [0.88296761 0.47455471]

recall:
 [0.80361388 0.62479062]

f-score:
 [0.84142395 0.53940709]

              precision    recall  f1-score   support

           0       0.88      0.80      0.84      2103
           1       0.47      0.62      0.54       597

    accuracy                           0.76      2700
   macro avg       0.68      0.71      0.69      2700
weighted avg       0.79      0.76      0.77      2700
```

Figure 3: *classification report* del modello 3 (undersampling)