

## Assignment 2 Ferrario Gabriele 817518

Durante l'analisi del dataset mi sono accorto della presenza di valori scorretti nei campi EDUCATION e MARRIAGE ed ho provveduto alla loro eliminazione (nel X\_test e nel X\_train). Proseguendo con l'analisi mi sono accorto di dover affrontare un task di classificazione in cui la variabile target é sbilanciata poiché sono presenti 18626 istanze della classe 0 e solo 5319 istanze appartenenti alla classe 1. Ho regolato la distribuzione delle classi tramite l'undersampling, selezionando 5319 istanze casuali della classe di maggioranza (classe 0) con l'obiettivo di avere lo stesso numero di istanze per ogni classe. Ho scelto di effettuare l'undersampling per non ottenere in fase di addestramento un modello che tende maggiormente verso la classe di maggioranza, non ho effettuato l'oversampling poiché questa tecnica potrebbe far allontanare il modello dalla realtà osservata (perché richiede la creazione di molte istanze per bilanciare il dataset). Ho verificato che non ci siano valori mancanti (nan) ed ho trasformato le features ridimensionandole in un intervallo  $[0, 1]$  tramite il MinMaxScaler<sup>1</sup> (ho utilizzato questo scaler poiché in fase di valutazione del modello tra quelli utilizzati é stato quello che mi ha fornito le performance migliori).

Per l'allenamento e la valutazione del modello ho deciso di suddividere i dati a disposizione per il training in 90% training e 10% validation.

Il modello é una rete neurale composta da 4 layer:

1. layer di input che accetta vettori di dimensione pari a 23 (dove 23 é il numero di features utilizzate per la predizione, escludendo il campo ID);
2. hidden layer denso con un numero di unità pari a 32 e come funzione di attivazione usa la relu;
3. hidden layer denso con un numero di unità pari a 16 e come funzione di attivazione usa la relu;
4. output layer denso con un numero di unità pari a 1 e come funzione di attivazione usa la funzione sigmoid poiché il task é una classificazione binaria (0 o 1);

---

<sup>1</sup>Questo estimator scala e traduce ogni features individualmente in modo tale che si trovi nell'intervallo dato sul set di addestramento, ad esempio tra zero e uno.

Ho usato questa configurazione della rete poiché nelle prove che ho fatto è stata quella che si è comportata meglio ottenendo le performance migliori. Come optimizer ho utilizzato adam mentre come loss function la binary\_crossentropy (funzione di perdita usata nei tasks di classificazione binaria). Come metriche ho usato l'accuracy e F1 score.

Il modello è stato allenato tramite 100 epoche e con la dimensione dei batch pari a 16. Con il passare delle epoche i valori delle metriche si alzano fino a raggiungere cifre vicine a 0.73 per l'accuratezza e di 0.69 per F1.

Nota: questo modello è stato utilizzato come base in tutti i test descritti successivamente.

Ho usate varie tecniche di regolarizzazione, tra cui:

- **L1**: applicando una regolarizzazione al kernel dei layer con un regularization factor di 0.01. Questa regolarizzazione mi ha portato ad ottenere un'accuratezza di circa 0.5;
- **L2**: applicando una regolarizzazione al kernel dei layer con un regularization factor di 0.01. Questa regolarizzazione mi ha portato ad ottenere un'accuratezza di circa 0.7;
- **Dropout**: utilizzando un rate pari a 0.2 (frazione di unità da abbandonare) su ogni strato nascosto. Questa regolarizzazione mi ha portato ad ottenere un'accuratezza di circa 0.71;
- **Early Stopping**: settando come quantità da monitorare la loss function e con un parametro di patience pari a 3 (numero di epoche senza alcun miglioramento dopo le quali l'allenamento verrà interrotto). Questa regolarizzazione mi ha portato ad ottenere un'accuratezza di circa 0.72;

Analizzando i pesi di un neurone ho notato che tutte le tecniche di regolarizzazione hanno diminuito i pesi eccetto il dropout nel quale sono aumentati leggermente (circa di 0.02). La regolarizzazione che ha generato i pesi minori è stata la L1, mentre quella che ha avuto le performance migliori è stata l'early stopping.

Nota: la regolarizzazione non è stata applicata sul bias poiché di solito richiede meno dati per essere adattato.

Inoltre, ho fatto un test effettuando una selezione delle features basandomi sulla matrice di correlazione, selezionando le features più correlate alla target. Il modello utilizzato in questo test é composto dallo stesso numero di neuroni, layer e stessa configurazione di quello definito precedentemente. In fase di allenamento ho utilizzato l'early stopping perché precedentemente ha permesso di ottenere i migliori valori di performance. Purtroppo questo test non ha migliorato il risultato fornendo un'accuratezza pari a 0.71 e uno score F1 pari a 0.66

Infine, per effettuare le predizioni sul test ho utilizzato il modello che utilizza tutte le features ottenuto tramite early stopping, perché é stato quello che si é comportato meglio.