**ADVANCE MACHINE LEARNING ASSIGNMENT**

Davide Brinati – Matricola 771458 – Data Science

Il dataset oggetto della nostra analisi, risulta essere composto da 2 parti:

* *Training*: 27000 osservazioni e 24 colonne
* *Testing*: 3000 osservazioni e 23 colonne

Tale dataset fornisce informazioni relative a diversi clienti e la loro capacità di adempire il pagamento il mese successivo. Questa informazione è fornita dalla variabile *default.payment.next.month*, la quale risulta essere binaria ed assume i seguenti valori: 0 se il pagamento del cliente nel mese successivo non va in default e 1 nel caso opposto. Default*.payment.next.month* è la variabile target.

**Preprocessing**

Nella fase esplorativa del dataset è emerso un problema di “class-imbalance” della nostra variabile target, come si può osservare dal grafico sottostante.

Immagine che contiene screenshot

Descrizione generata con affidabilità elevata

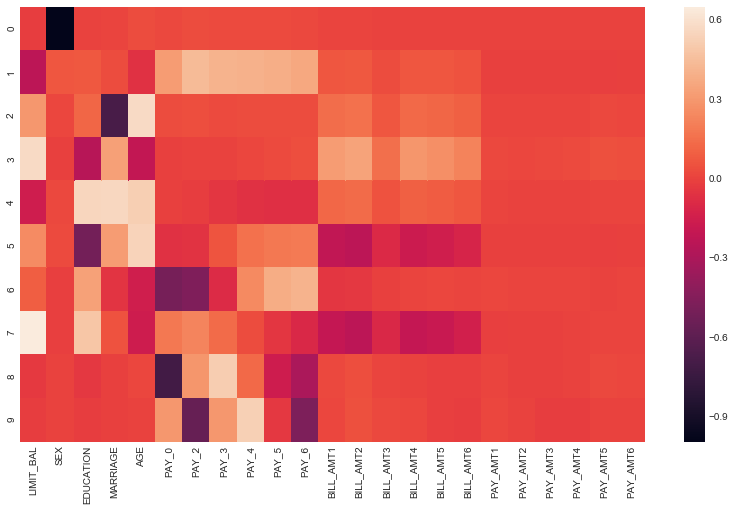
Le osservazioni aventi come target il valore positivo, risultano essere poco più di 5996, pari al 22% del totale. Si è optato quindi per un “oversampling”, ovvero un ricampionamento della classe rara, in modo da fornire all’algoritmo più casi nei quali la variabile target risulta essere pari ad 1. Si è scelto di ricampionare la classe rara 21004 volte, ed equiparare i casi in cui il target assume il valore 0 e i casi in cui assume 1.

Dopo l’operazione di oversampling, la variabile target risulta essere così distribuita:

Immagine che contiene screenshot

Descrizione generata con affidabilità elevata

Nell’ultima parte di preprocessing si è effettuata una riduzione della dimensione del dataset tramite PCA, diminuendo il numero di feature da 23 a 10. Di seguito è possibile osservare la matrice di correlazione fra le 10 features create e le 23 features di partenza. Nella fase di modelling verranno paragonati i risultati del modello con tutte le variabili e quelli del modello avente per input il dataframe sottoposto a PCA.



**Modello**

Il train è stato diviso in train e validation, in modo da allenare il modello sul train e testarlo poi sul validation. La rete neurale creata risulta avere i seguenti layer, di dimensione:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Layer (type) Output Shape Param # ================================================================= dense\_61 (Dense) (None, 1000) 24000 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ dense\_62 (Dense) (None, 400) 400400 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ dropout\_11 (Dropout) (None, 400) 0 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ dense\_63 (Dense) (None, 100) 40100 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ dense\_64 (Dense) (None, 90) 9090 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ dense\_65 (Dense) (None, 80) 7280 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ dense\_66 (Dense) (None, 60) 4860 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ dropout\_12 (Dropout) (None, 60) 0 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ dense\_67 (Dense) (None, 40) 2440 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ dense\_68 (Dense) (None, 30) 1230 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ dense\_69 (Dense) (None, 20) 620 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ dense\_70 (Dense) (None, 10) 210 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ dense\_71 (Dense) (None, 5) 55 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ dense\_72 (Dense) (None, 2) 12 ================================================================= Total params: 490,297 Trainable params: 490,297 Non-trainable params: 0 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ogni layer ha come funzione di attivazione la *relu*, ad esclusione dello strato di output che ha la *sigmoid* come funzione di attivazione, dato che il nostro target risulta essere binario. La funzione di loss è la *binary\_crossentropy* e l’ottimizzatore è *adam*. Sono stati aggiunti strati di dropout per ridurre l’overfitting.

Tale modello è stato testato prima sul dataframe con tutte le 23 feature e successivamente sul dataframe sottoposto a PCA, con 10 features. In entrambi casi sono state utilizzate 50 epoche.

Immagine che contiene testo, mappa

Descrizione generata con affidabilità molto elevata

Risulta chiaro che dando in input il dataframe sottoposta a PCA non si hanno buoni risultati sul validation, pertanto tale strada verrà abbandonata. Utilizzeremo quindi il modello che prende in input tutte le variabili, il quale ha portato ai seguenti risultati sul validation.

Immagine che contiene testo

Descrizione generata con affidabilità molto elevata

Immagine che contiene testo, mappa

Descrizione generata con affidabilità molto elevata

Infine, ho addestrato il modello descritto sopra sull’intero train, prima di effettuare la predizione del test.