# Advanced Machine Learning

Primo Assignment

Simone Paolo Mottadelli, 820786

# Contents

# 1 Data processing

Il dataset in oggetto presenta 33884 istanze relative agli appartamenti in affitto a New York su AirBnB nel 2019. Tra i 10 attributi presenti nel dataset, l'attributo *price* è la variabile target ed è di tipo numerico. Infatti, il problema che si vuole risolvere è un problema di regressione, dove si vuole predire il prezzo degli appartamenti, o stanze private, in base al valore delle altre variabili esplicative. La prima cosa che si può notare guardando il dataset è che alcuni prezzi di appartamenti e stanze private sono troppo elevati oppure hanno un prezzo nullo, dunque, queste osservazioni si possono considerare come degli outliers. Perciò, ho deciso di rimuovere dal dataset tutte le istanze avente un prezzo maggiore del quantile di ordine 0.95 oppure pari a 0 dollari, eliminando, dunque, 1410 istanze (4.16% del dataset).

La Figura 1 mostra che le variabili hanno una distribuzione "molto piccata"

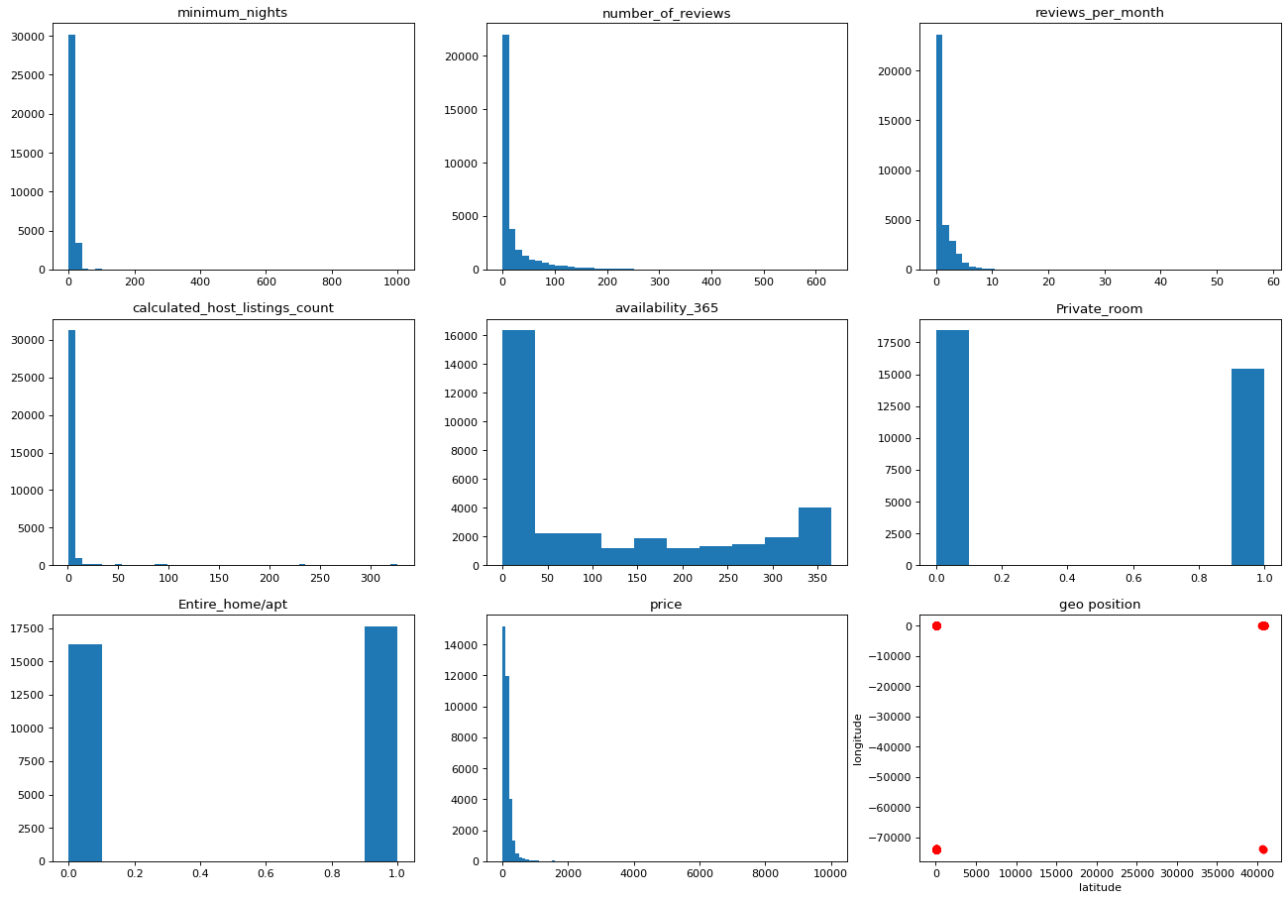


Figure 1: Variable distributions before the data processing phase

sulla parte sinistra del grafico. Ho cercato di rendere le distribuzioni delle variabili migliori andando ad applicare delle trasformazioni logaritmiche ($y =$

$log(x+1)$) alle seguenti variabili:

- *minimum_nights*;
- *number_of_reviews*;
- *reviews_per_month*;
- *calculated_host_listings_count*;
- *availability_365*.

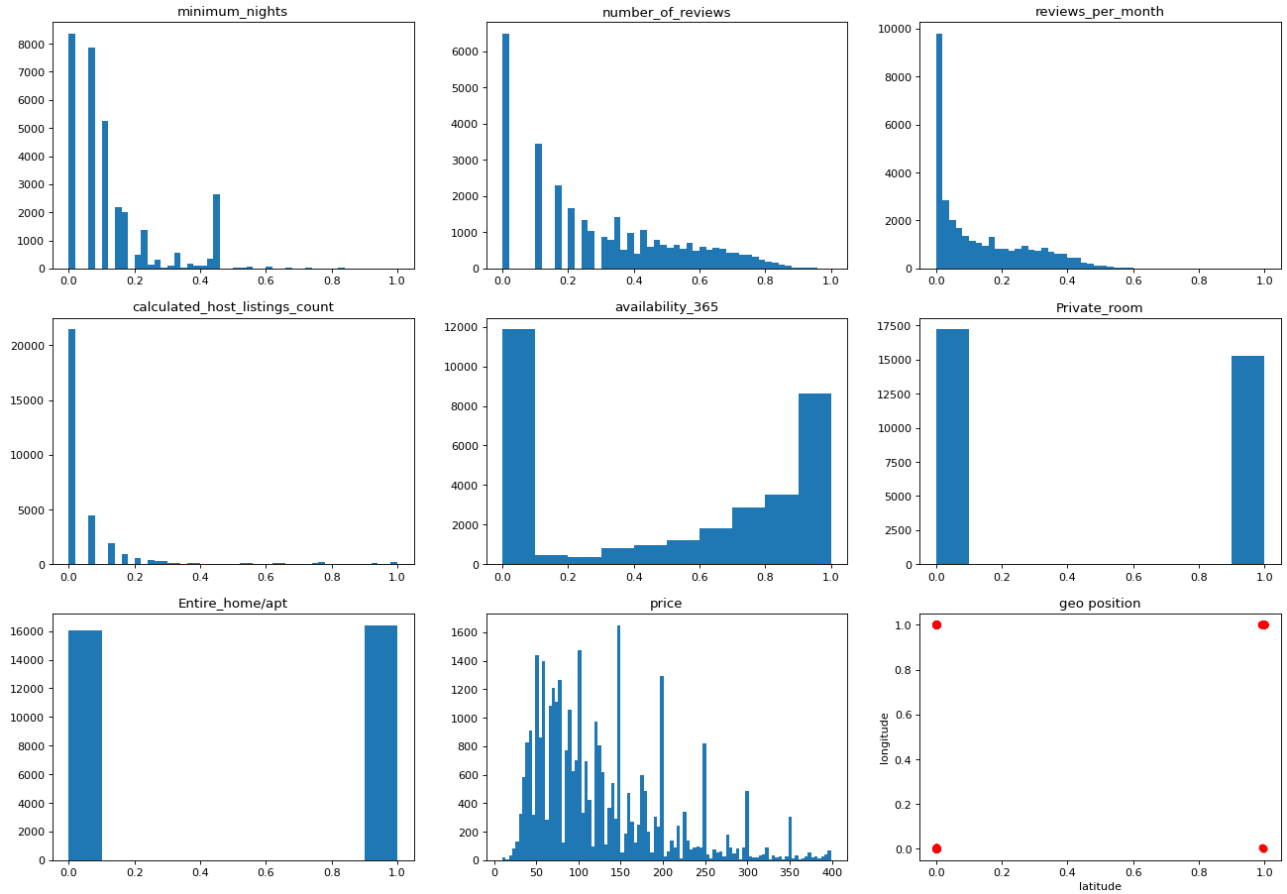


Figure 2: Variable distributions after the data processing phase

Inoltre, ho applicato una normalizzazione tra 0 ed 1 per le seguenti variabili:

- *minimum_nights*;
- *number_of_reviews*;
- *reviews_per_month*;
- *availability_365*;

- *longitude*;
- *latitude*;
- *calculated_host_listings_count*.

Le variabili *price*, *Entire_home/apt* e *Private_room* non sono state modificate. La Figura 2 mostra le distribuzioni delle variabili dopo la fase di data processing.

## 2 Regression

Per risolvere il problema di regressione ho deciso di utilizzare una feed forward neural network. Per configurare il numero di strati nascosti e di neuroni nascosti ho effettuato diverse prove, ma non sembrava ci fosse una configurazione nettamente superiore ad un'altra. Semplicemente, ho deciso di utilizzare 4 strati nascosti perchè mi sembrava un giusto compromesso tra complessità del modello e capacità di generalizzazione. Per quanto riguarda il numero di neuroni nascosti, ho deciso di seguire le linee guida descritte nel paper [1], in cui si afferma che il numero di neuroni nascosti dovrebbe essere minore del doppio della grandezza dello strato di input, quindi, siccome lo strato di input ha 9 neuroni, il numero di neuroni nascosti dovrebbe essere inferiore a 18. Pertanto, ho utilizzato una rete neurale con 4 layers nascosti, ciascuno costituito da 14 neuroni.
L'output function che ho deciso di utilizzare è la *relu*, dal momento che i prezzi vanno da 0 a infinito, mentre la scelta della loss function è ricaduta sulla *mean squared error* perchè penalizza maggiormente gli errori grandi commessi dalla rete.
Come ottimizzatore, ho deciso di utilizzare Adam perchè è stato progettato specificatamente per le deep neural networks ed è noto per essere molto performante.
Ho provato a valutare le performance dei modelli utilizzando diverse activation function. In particolare:

- relu
- elu
- leaky relu con $\alpha \in \{0.3, 0.2, 0.1\}$
- selu

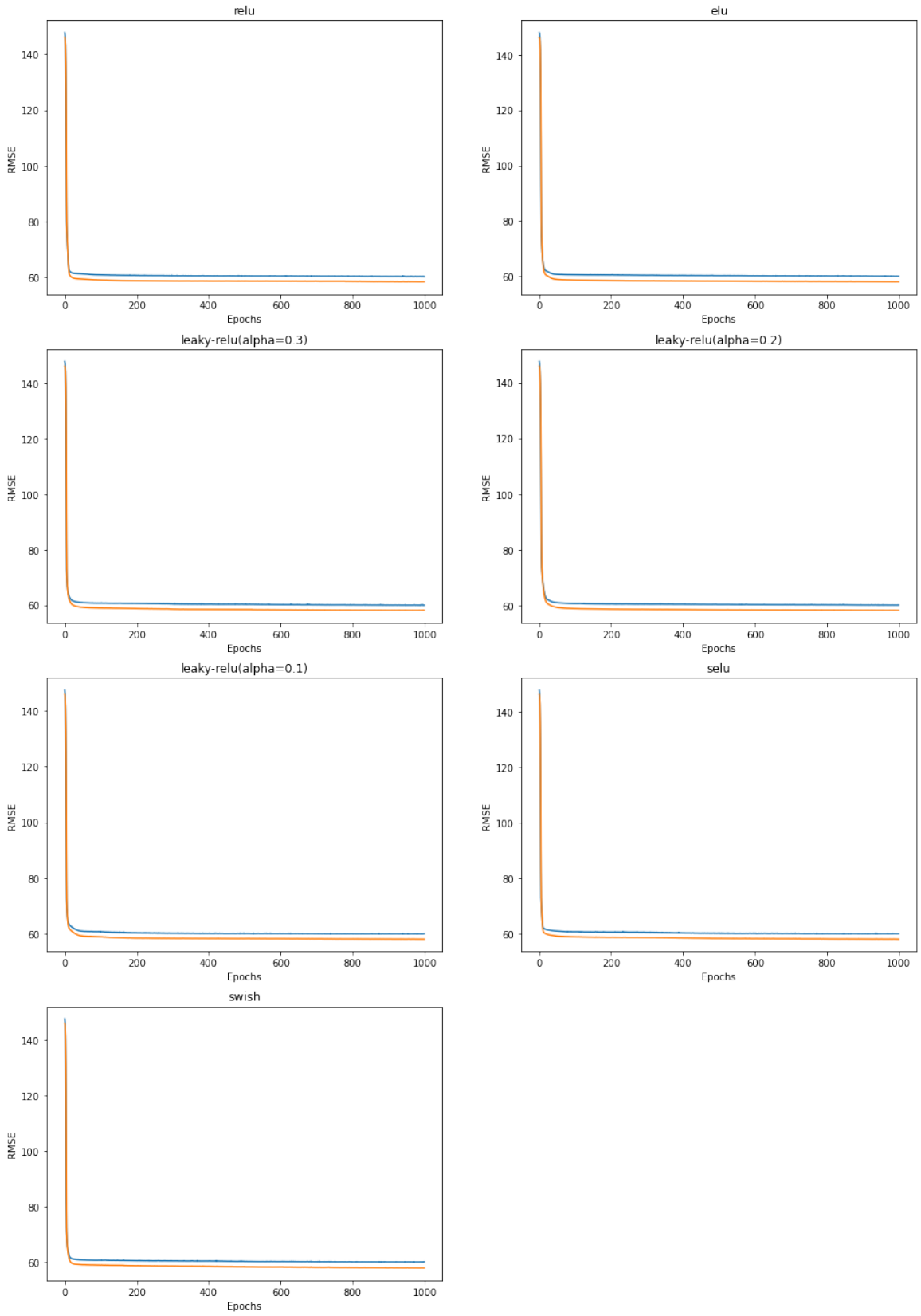


Figure 3: Training error (yellow curve) vs validation error (blue curve)

- swish

Come si può osservare dai risultati in figura 3, non esiste una activation function migliore e, per questo motivo, ho deciso di utilizzare la *relu* poichè è l'activation function più utilizzata al giorno d'oggi.
Il *root mean squared error* calcolato sul test set è circa 60 e questo risultato, a mio avviso, non sembra essere buono, visto che stiamo parlando di prezzi per soggiornare in delle case e stanze private. Tuttavia, anche facendo diverse prove cambiando il numero di layer, numero di neuroni per layer, ottimizzatore, non sono riuscito a migliorare in modo netto i risultati ottenuti.

## References

[1] F. Panchal and Mahesh Panchal. Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture. *International Journal of Computer Science and Mobile Computing*, 3:455–464, 01 2014.