Alma Mater Studiorum · Università di Bologna

Scuola di Scienze
Dipartimento di Fisica e Astronomia
Corso di Laurea in Fisica

Previsione dell'età biologica mediante segnali PPG

Relatore:
Prof. Gastone Castellani

Correlatore:
Dott. Nico Curti

Presentata da:
Andrea Bartolucci

Anno Accademico 2018/2019

Sommario

La fotopletismografia (PPG) è una tecnica recente per la misura del volume del sangue arterioso basata sullo studio della variazione dell'assorbanza fra emoglobina ossigenata e deossigenata. I suoi vantaggi sono la semplicità dell'apparato, il basso costo e la non invasività. L'età biologica di un individuo è determinata non solo dall'età cronologica, ma anche da altri fattori come lo stile di vita, possibili patologie, dieta, esercizio fisico, stress. Lo studio dell'età biologica è quindi importante per la possibilità di diagnosticare possibili malattie o problemi con il proprio stile di vita. In questa tesi svilupperemo diversi modelli di machine learning e reti neurali per la regressione dell'età biologica utilizzando delle features estratte da un dataset di segnali PPG. La quantità di sangue che transita nel sistema vascolare dipende infatti dal tono vascolare e questo a sua volta dipende dall'età del soggetto.

Indice

Capitolo 1

Fotopletismografia (PPG)

1.1 Principi

La fotopletismografia (PPG) è un importante metodo non invasivo per la misura del volume del sangue arterioso basata sullo studio dell'assorbanza dei tessuti. In particolare sfrutta il fatto che i gruppi eme dell'emoglobina (fig. 1.1) sono in grado di assorbire una lunghezza d'onda diversa da quella assorbita tipicamente dagli altri tessuti organici, in particolare queste sono principalmente 660 nm per la deossiemoglobina (emoglobina non legata ad una molecola di ossigeno) e di 940 nm per la ossiemoglobina (emoglobina legata ad una molecola di ossigeno) [1]. Grazie alla sua semplicità e al suo basso costo, la PPG offre un metodo veloce per la possibile diagnosticazione di problemi cardiovascolari.

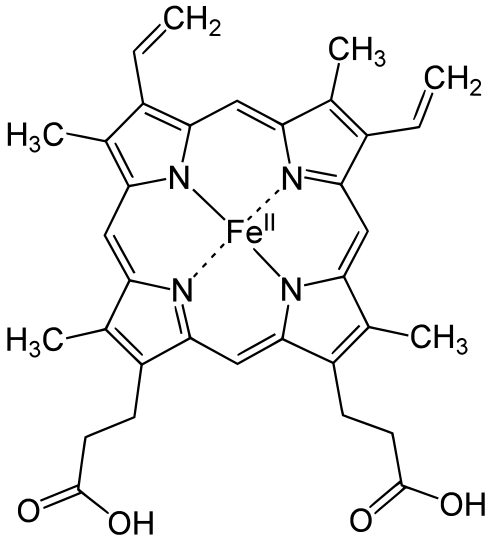

Figura 1.1: Gruppo eme. Le molecole d'ossigeno si legano all'atomo di Fe.

1.2 Apparato e fuzionamento

Un apparato (mostrato in fig. 1.2) per la PPG è composto da un LED con una lunghezza d'onda vicina a quella di assorbimento dell'emoglobina e da un fotosensore. L'apparato emette una luce con intensità I_0 e misura l'intensità residua I_1, cioè la luce non assorbita dai tessuti e dai vasi sanguigni arteriosi. Definiamo l'assorbanza come

$$A = \ln\left(\frac{I_0}{I_1}\right). \tag{1.1}$$

Possiamo ora collegare l'assorbanza di un tessuto alla lughezza del cammino del raggio di luce utilizzando la legge di Beer-Lambert

$$A = \epsilon_\lambda l M, \tag{1.2}$$

dove ϵ_λ è il coefficiente di assorbimento molare, M è la molarità della soluzione e l è il cammino geometrico. L'assorbanza così calcolata corrisponde alla somma dell'assorbanza arteriosa A_A e dell'assorbanza degli altri tessuti A_{altri} [2]. Calcolando la derivata dell'assorbanza totale rispetto al tempo otteniamo

$$\frac{dA}{dt} = \frac{dA_A}{dt} + \frac{dA_{\text{altri}}}{dt} \tag{1.3}$$

$$= \epsilon_{\lambda A} M_A \frac{dl_A}{dt} + \epsilon_{\lambda \text{altri}} M_{\text{altri}} \frac{dl_{\text{altri}}}{dt} \tag{1.4}$$

$$\approx \epsilon_{\lambda A} M_A \frac{dl_A}{dt}, \tag{1.5}$$

dove abbiamo utilizzato il fatto che il volume di emoglobina dei tessuti non arteriosi è approssimativamente costante [3]. Vediamo quindi che possiamo dividere l'assorbanza totale in una parte che varia proporzionalmente alla larghezza delle arterie l_A e una parte costante che possiamo rimuovere con un filtro appropriato in modo da poter studiare isolatamente il flusso sanguigno nelle arterie. Questo risultato è di grande importanza dato che permette di collegare le misure del fotosensore con la il volume dei vasi sanguigni e quindi indirettamente con parametri vitali quali la pressione sanguigna e la frequenza cardiaca.

1.3 Caratteristiche principali di un segnale PPG

Un segnale PPG è caratterizzato da una serie di impulsi che si ripetono periodicamente. Il singolo impulso è a sua volta caratterizzato da una prima parte di salita rapida fino ad un picco chiamato picco sistolico, seguita da una parte di lenta discesa nella quale si può presentare un picco chiamato picco diastolico (fig. 1.3) [4]. La prima parte di salita è

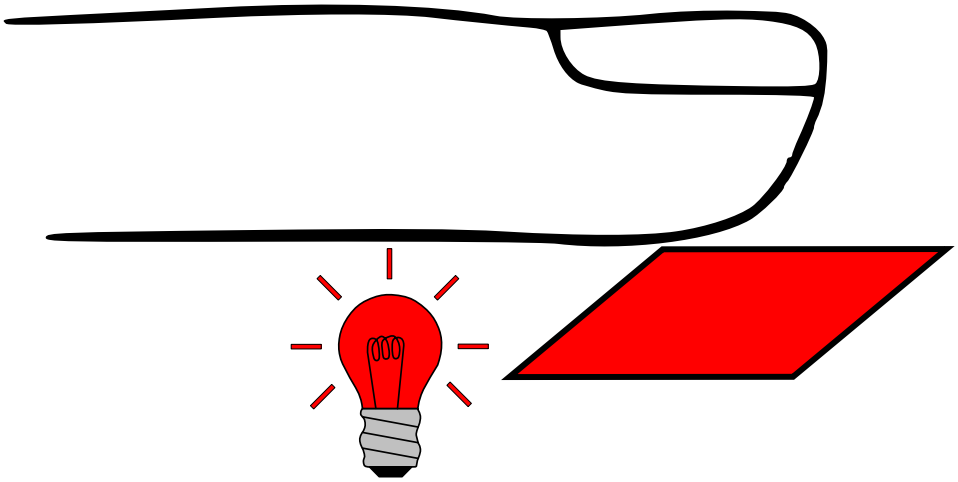

Figura 1.2: Schema dell'apparato. Il dito viene illuminato da una luce led e un sensore rileva la luminosità.

associata alla fase sistolica del ciclo cardiaco, infatti durante questa fase la valvola aortica si apre e la pressione all'interno dei vasi sanguigni aumenta di conseguenza. La parte di discesa è associata alla fase diastolica, nella quale il cuore si rilassa e la pressione sanguigna diminuisce. Il picco diastolico è associato alla riflessione dell'onda di pressione che parte dall'arteria succlavia e si riflette alla periferia del sistema circolatorio. Quest'ultimo è di particolare importanza perché può essere utilizzato come indicatore della rigidità delle arterie, fattore correlato con l'età biologica.

1.4 Modello di windkessel

Per spiegare quantitativamente la presenza del picco diastolico esiste un modello sviluppato inizialmente dal fisiologo Otto Frank e poi migliorato in tempi più recenti, chiamato modello di windkessel [5]. Il modello consiste in un'analogia con un circuito elettrico composto da componenti passive, nel quale la corrente corrisponde al flusso sanguigno, la differenza di potenziale alla pressione sanguigna e le caratteristiche delle arterie sono rappresentate dalle componenti del circuito. Il circuito è quello mostrato in fig. 1.4. La funzione della corrente si ricava quindi risolvendo la seguente equazione differenziale

$$\left(1+\frac{R_1}{R_2}\right)I(t)+\left(R_1C+\frac{L}{R_2}\right)\frac{dI(t)}{dt}+LC\frac{d^2I(t)}{dt^2}=\frac{P(t)}{R_2}+C\frac{dP(t)}{dt}, \tag{1.6}$$

utilizzando come funzione per la corrente

$$I(t)=\begin{cases} I_0\sin\left(\frac{\pi\mathrm{mod}(t,T)}{T_s}\right), & \text{se } 0\leq \mathrm{mod}(t,T)<T_s \\ 0, & \text{se } T_s\leq \mathrm{mod}(t,T)<T \end{cases} \tag{1.7}$$

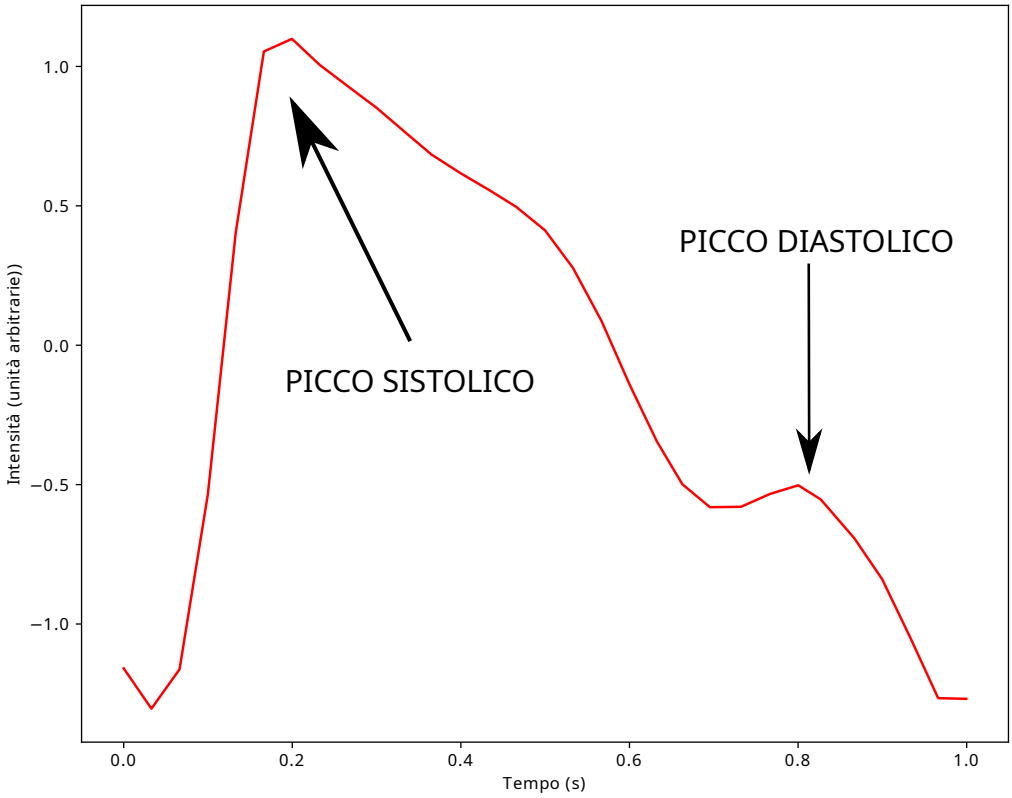

Figura 1.3: Esempio di impulso ppg.

dove T_s è la durata del ciclo sistolico e $T > T_s$ è la durata del singolo impulso. Un esempio di risoluzione dell'eq. 1.6 è in fig 1.5.

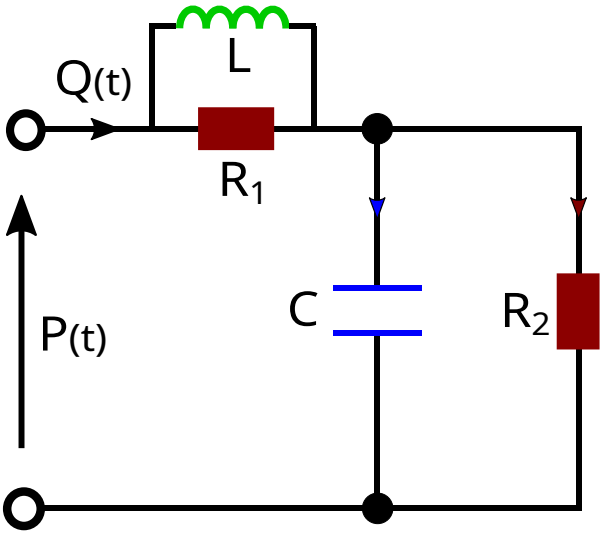

Figura 1.4: Schema elettrico del circuito di windkessel.

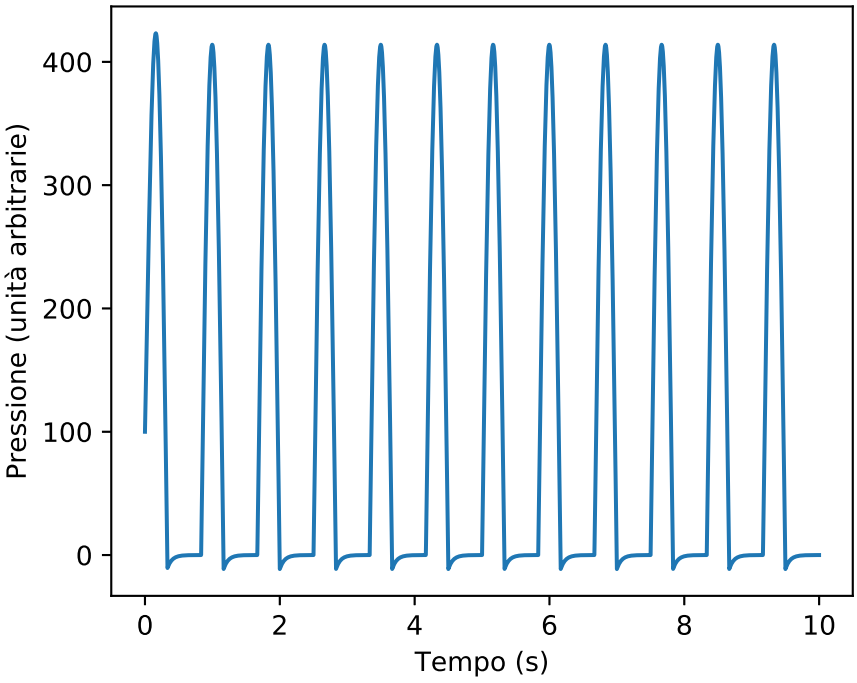

Figura 1.5: Soluzione dell'equazione differenziale di windkessel.

1.5 Studio della derivata seconda

Una parte importante nello studio dei segnali PPG è lo studio della derivata seconda. Come si può vedere dalla fig. 1.6, la derivata seconda di un impulso (chiamata anche sdppg, **s**econd **d**erivative of **ppg**) presenta solitamente 5 picchi, chiamati a,b,c,d,e. Alcune combinazioni delle altezze di questi picchi è già stata studiata in precedenza:

Rapporto b/a [6] ha dimostrato che questa quantità è correlata alla rigidità delle arterie e aumenta con l'età;

Rapporto c/a [6] ha dimostrato che questa quantità è correlata negativamente alla rigidità delle arterie e diminuisce aumenta con l'età;

(b-c-d-e)/a [6], chiamato anche AGI ha dimostrato che questa quantità incrementa con l'età e può essere utile per lo screening dell'aterosclerosi.

Noi proveremo ad utilizzare anche altre combinazioni dei tempi e delle ampiezze di questi picchi. Una lista completa si trova nel paragrafo 3.3.1.

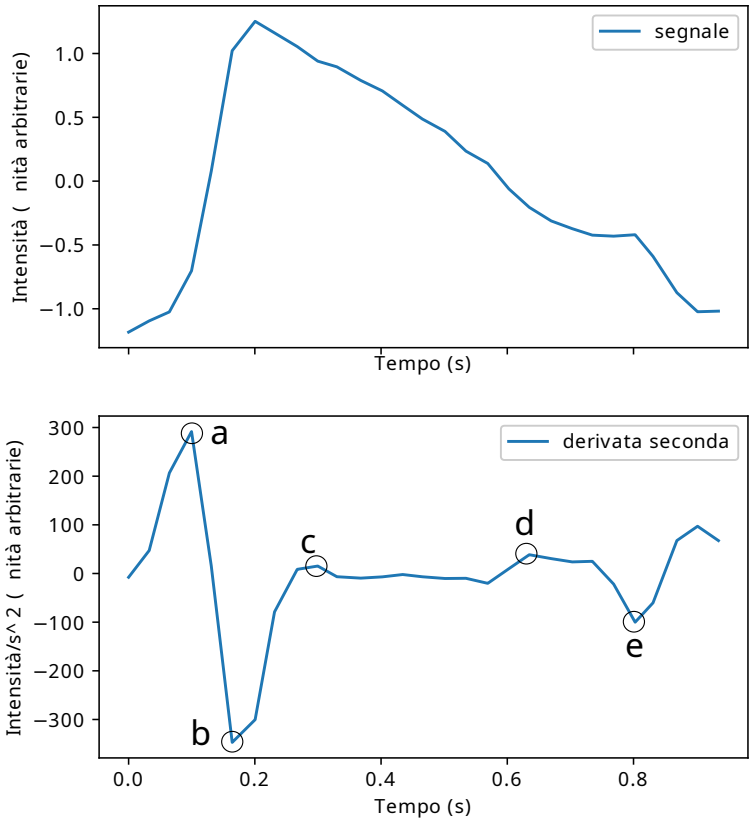

Figura 1.6: Grafico della derivata seconda con punti a,b,c,d,e.

Capitolo 2

Machine Learning

2.1 Introduzione

In questo capitolo introdurremo i metodi di machine learning utilizzati in questo studio. Partiremo dai principali metodi di regressione lineare, poi parleremo dei metodi generalmente utilizzati per preprocessare per rendere le regressioni numericamente più stabili. Introdurremo inoltre i concetti di bias e varianza e i principali metodi per evitare l'overfitting. Studieremo l'algoritmo di cross-validation per valutare il funzionamento di un modello e i principali indici di bontà del fit. Infine verranno introdotte le reti neurali.

2.2 Problemi di regressione

In un problema di regressione sono note le caratteristiche numeriche $\mathbf{x}_i$ di un qualche fenomeno e, per ognuna di esse, un valore reale associato y_i. Lo scopo della regressione è quello di stimare la relazione che intercorre tra le osservazioni e le caratteristiche del fenomeno, tenendo conto del fatto che le osservazioni sono soggette ad un rumore indipendente dalle altre variabili. Normalmente abbiamo un set di dati di training $(\mathbf{x}_1, y_1) \dots (\mathbf{x}_N, y_N)$, dove $\mathbf{x}_j$ è un vettore di lunghezza p contenente le features e y_j sono le variabili dipendenti associate ai vettori $\mathbf{x}_j$. Le prestazioni del modello vengono poi valutate su un set separato chiamato set di testing.

2.3 Least square fitting

Prendiamo un modello del tipo

$$f(\mathbf{x}) = \beta_0 + \sum_{j=1}^{p} x_j \beta_j. \tag{2.1}$$

L'obbiettivo del LSF è quello di trovare i parametri $\beta = (\beta_0, \ldots, \beta_p)$ che minimizzano [7, cap. 3.2]

$$RSS(\beta) = \sum_{i=1}^{N} \left(y_i - f\left(\mathbf{x}_i\right)\right)^2 \tag{2.2}$$

$$= \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 . \tag{2.3}$$

Definiamo ora $\mathbf{X}$ come la matrice $N \times (p+1)$ in cui ogni riga corrisponde un vettore di features di un singolo input (con un 1 sulla prima colonna). Definiamo inoltre $\mathbf{y}$ come il vettore di lunghezza N che contiene le variabili dipendenti. Notiamo che possiamo riscrivere la (2.3) come

$$RSS(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta). \tag{2.4}$$

Differenziando rispetto a β otteniamo

$$\frac{dRSS}{d\beta} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta). \tag{2.5}$$

Imponendo che questa derivata sia zero otteniamo

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0, \tag{2.6}$$

da cui otteniamo la soluzione

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\mathbf{y}. \tag{2.7}$$

Inserendo la (2.7) nella (2.1) otteniamo che i valori predetti $\hat{\mathbf{y}}$ per il set di training $\mathbf{X}$ sono

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^\mathbf{T}\mathbf{y}. \tag{2.8}$$

Una importante limitazione di questo metodo è data dalla possibilità che la matrice $\mathbf{X}^T\mathbf{X}$ sia singolare, vediamo infatti che in questo caso i coefficienti dell'eq. 2.7 non sono ben definiti. Questo caso avviene quando il numero di features supera il numero di samples o quando delle features non sono linearmente indipendenti.

2.4 Overfitting e underfitting

2.4.1 Bias e varianza

Vogliamo ora studiare da un punto di vista teorico l'errore previsto di un fit di regressione $\hat{f}(\mathbf{X})$ quando valutato con un set di test preso da una variabile casuale $\mathbf{y} = f(\mathbf{x}) + \epsilon$, con

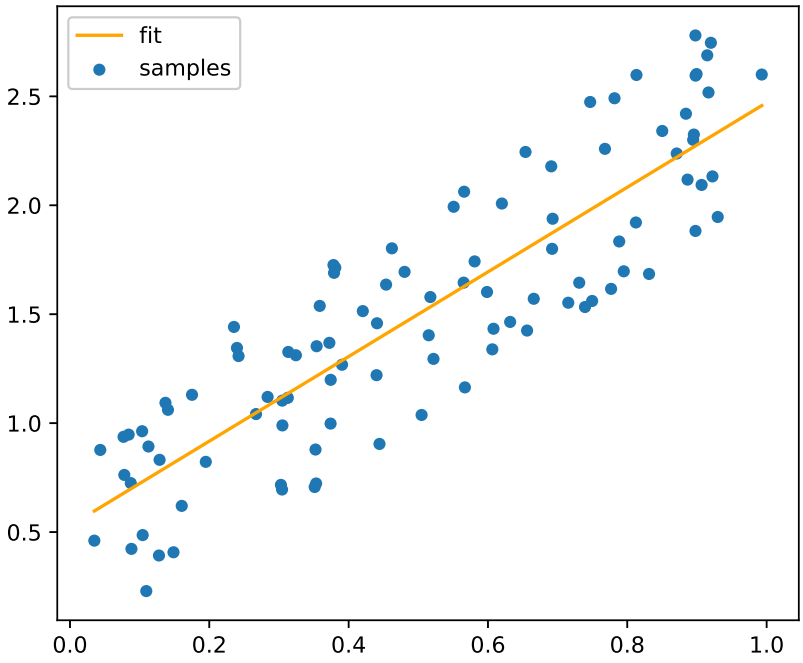

Figura 2.1: Esempio di regressione lineare.

$\mathbf{x}$ fisso e le componenti di ϵ sono variabili aleatorie indipendenti con media 0 e varianza σ_ϵ^2. Da notare che $\hat{f}$ viene calcolata utilizzando un set di training diverso da quello di test. Il valore atteso dell'errore quadratico medio è

$$\begin{aligned}
\text{Err} &= E\left[\left(\mathbf{y} - \hat{f}(\mathbf{x})\right)^2\right] \\
&= E\left[\left(f(\mathbf{x}) + \epsilon - \hat{f}(\mathbf{x})\right)^2\right] \\
&= \sigma_\epsilon^2 + E\left[\left(f(\mathbf{x}) - \hat{f}(\mathbf{x})\right)^2\right] + 2E\left[\epsilon\left(f(\mathbf{x}) - \hat{f}(\mathbf{x})\right)\right] \\
&= \sigma_\epsilon^2 + E\left[\left(f(\mathbf{x}) - E[\hat{f}(\mathbf{x})] + E[\hat{f}(\mathbf{x})] - \hat{f}(\mathbf{x})\right)^2\right] \\
&= \sigma_\epsilon^2 + \left(f(\mathbf{x}) - E[\hat{f}(\mathbf{x})]\right)^2 + E\left[\left(\hat{f}(\mathbf{x}) - E[\hat{f}(\mathbf{x})]\right)^2\right] \\
&= \sigma_\epsilon^2 + \text{Bias}^2\left(\hat{f}(\mathbf{x})\right) + \text{Var}^2(\hat{f}(\mathbf{x})),
\end{aligned} \tag{2.9}$$

dove il terzo termine della terza equazione è 0 perché $\hat{f}(\mathbf{x})$ è stata trovata utilizzando un set indipendente da $\mathbf{y}$ e il valore medio di ϵ è 0. Vediamo quindi dalla (2.9) che l'errore atteso si può dividere in tre parti: la prima è data solamente dall'errore sulle osservazioni e quindi non può essere evitata, la seconda è il quadrato della distanza tra la media della stima e la media vera mentre la terza è la varianza (diversa da 0 poiché la stima dipende dal possibile errore sui dati di training). Come esempio, vediamo come si scompone l'errore atteso di un LSF:

$$\begin{aligned}
\text{Err} &= \sigma_\epsilon^2 + \text{Bias}^2\left(\hat{f}(\mathbf{x})\right) + E\left[\left(\hat{f}(\mathbf{x}) - E[\hat{f}(\mathbf{x})]\right)^2\right] \\
&= \sigma_\epsilon^2 + \text{Bias}^2\left(\hat{f}(\mathbf{x})\right) + E\left[\left(\mathbf{x}^T\mathbf{A}\mathbf{y} - E[\hat{f}(\mathbf{x})]\right)^2\right] \\
&= \sigma_\epsilon^2 + \text{Bias}^2\left(\hat{f}(\mathbf{x})\right) + E\left[\left(\mathbf{x}^T\mathbf{A}\epsilon\right)^2\right] \\
&= \sigma_\epsilon^2 + \text{Bias}^2\left(\hat{f}(\mathbf{x})\right) + \left|\mathbf{A}^T\mathbf{x}\right|^2\sigma_\epsilon^2, \qquad (2.10)
\end{aligned}$$

dove $\mathbf{A} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$. Normalmente all'aumentare della complessità del modello diminuisce il bias e aumenta la varianza. Un modello con molto bias può non catturare le relazioni esistenti tra le features e la variabile dipendente, questa situazione è chiamata underfitting. Al contrario, un modello con una varianza alta è molto sensibile alle fluttuazioni dei valori nel set di training e rischia di imparare il rumore dei dati casuali, aumentando l'errore nella fase di validazione (paragrafo 2.8). In fig. 2.2 possiamo vedere in modo qualitativo come variano bias e varianza al variare della complessità del modello utilizzato. Enunciamo ora un importante teorema sulla quantità di bias e varianza di un modello.

2.4.2 Teorema di Gauss-Markov

Il teorema di Gauss-Markov afferma che, limitandosi ai modelli lineari con bias nullo, il LSF è quello con una varianza minore. Una conseguenza di questo teorema è che c'è un limite fondamentale alla varianza e quindi all'errore atteso di un modello senza bias. Ciò non significa che il LSF è il modello con minor errore assoluto possiblile, infatti come vedremo nella sezione 2.5 si possono utilizzare modelli con un bias non nullo ma con una varianza molto più bassa.

2.5 Regressione lineare con penalità

2.5.1 Regressione ridge

La regressione di ridge impone una penalità sui coefficienti della regressione per evitare che crescano troppo. Procediamo come nel LSF, utilizzando però come errore [7, cap.

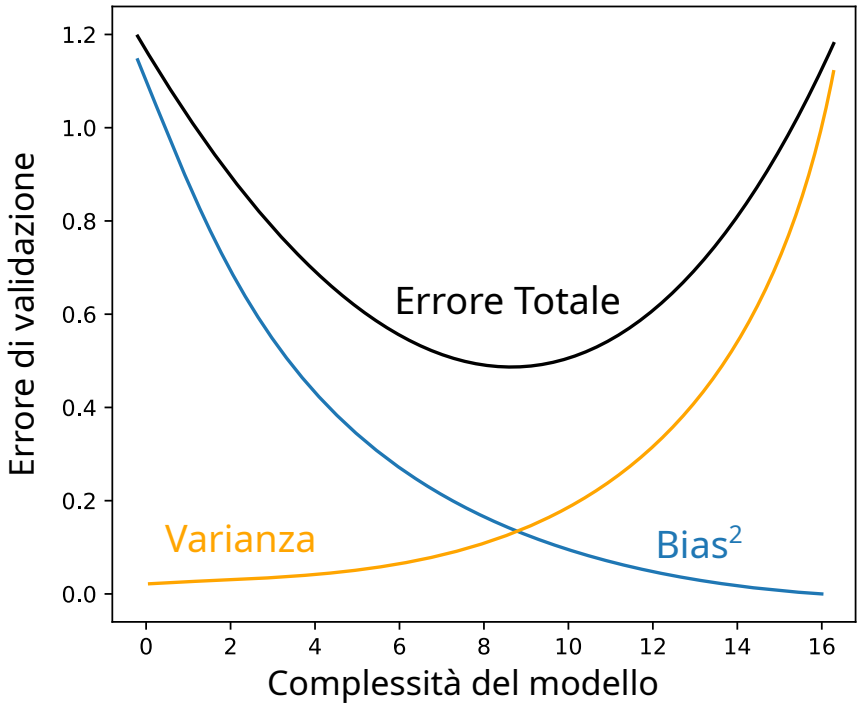

Figura 2.2: Immagine esplicativa nella quale si possono vedere gli effeti di bias e variaza al variare della complessità del modello.

3.4.1]

$$E_{ridge} = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda|\beta|^2, \tag{2.11}$$

dove $\lambda \geq 0$ è un parametro scelto a priori (più è alto, più i coefficienti sono schiacciati verso lo zero). L'obbiettivo è ancora una volta quello di trovare il vettore di coefficienti β che minimizza la (2.11). Possiamo verificare che la soluzione è data da

$$\hat{\beta}_{ridge} = \left(\mathbf{X}^T\mathbf{X} + \mathbf{I}\lambda\right)^{-1}\mathbf{X}^T\mathbf{y}, \tag{2.12}$$

dove $\mathbf{I}$ è la matrice identità $p \times p$. Notare che, al contrario della (2.7), la matrice a destra nella (2.12) è sempre invertibile. Il vettore $\mathbf{y}$ predetto da questo modello è quindi

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta}_{ridge} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X} + \mathbf{I}\lambda\right)^{-1}\mathbf{X}^T\mathbf{y}, \tag{2.13}$$

Possiamo usare la *singular value decomposition* (SVD) per capire meglio il funzionamento della regressione ridge: la SVD della matrice $\mathbf{X}$ è

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \tag{2.14}$$

dove U e V sono matrici ortogonali di dimensione rispettivamente $N \times p$ e $p \times p$ mentre D è una matrice diagonale i cui valori d_i sono tali per cui $d_1 \geq d_2 \geq \cdots \geq d_p \geq 0$ e sono chiamati valori singolari di $\mathbf{X}$. Utilizzando la scomposizione (2.14) nella (2.13) otteniamo

$$\hat{\mathbf{y}} = \mathbf{UD}\left(\mathbf{D}^T\mathbf{D} + \mathbf{I}\lambda\right)^{-1}\mathbf{DU}^T\mathbf{y} \tag{2.15}$$

$$= \sum_{k=1}^{p} \mathbf{u}_k \frac{d_k^2}{d_k^2 + \lambda}\mathbf{u}_k^T\mathbf{y}, \tag{2.16}$$

dove $\mathbf{u}_k$ sono le colonne di $\mathbf{U}$. Inoltre, dato che $\lambda \geq 0$, abbiamo che $0 \leq d_k^2/(d_k^2 + \lambda) \leq 1$. Vediamo che quindi la regressione ridge calcola le coordinate di $\mathbf{y}$ nella base ortonormale $\mathbf{U}$, scala le componenti di un fattore $d_k^2/(d_k^2+\lambda)$ (quindi rimpiccioilsce di più le componenti con d_k piccolo) e infine applica la trasformazione inversa di $\mathbf{U}$. Da notare che la regressione ridge si riduce alla LSF nel caso $\lambda = 0$ e che nella LSF i coefficienti di schiacciamento sono tutti pari a 1. I coefficienti δ_k sono chiamati valori singolari e come vedremo successivamente δ_k^2 rappresenta la varianza del dataset lungo il k-esimo asse principale della PCA.

2.5.2 Coordinate descent

Data: un vettore $\mathbf{x}$ iniziale
Result: un vettore $\mathbf{x}$ contenente le coordinate di un minimo di E
while *convergenza non raggiunta* **do**
  $i \leftarrow random(1, p)$;
  $x_i \leftarrow x_i - \alpha\frac{\partial E}{\partial x_i}(\mathbf{x})$;
end

Algoritmo 1: L'algoritmo del coordinate descent. α è una costante scelta a priori.

Introduciamo ora un algoritmo per la risoluzione di problemi di minimizzazione chiamato coordinate descent [8]. L'algoritmo si basa sull'idea che la minimizzazione di una funzione a più variabili si può ottenere spostandosi su una coordinata alla volta nella direzione in cui la funzione decresce, determinata dalla derivata parziale della funzione rispetto alla coordinata scelta $\frac{\partial E}{\partial x_i}$. Vediamo infatti che partendo da un vettore iniziale $\mathbf{x}^0 = (x_1^0, \ldots, x_p^0)$, possiamo definire il vettore $\mathbf{x}^{k+1}$ partendo dal vettore $\mathbf{x}^k$ scegliendo casualmente $i \in [1, \ldots, p]$ e definendo

$$x_j^{k+1} = \begin{cases} \underset{x_j^{k+1}}{arg\,min}\, E(x_1^k, \ldots, x_j^{k+1}, \ldots, x_p^k), & \text{se } i \neq j \\ x_j^k, & \text{altrimenti.} \end{cases}$$

Vediamo che ad ogni passaggio è garantito che

$$E(\mathbf{x}^0) \geq E(\mathbf{x}^1) \geq E(\mathbf{x}^2) \geq \dots. \tag{2.17}$$

Il processo non assicura una convergenza al minimo ma funziona nella maggior parte dei casi di interesse pratico. Un algoritmo che implementa questa idea è quello descritto in (alg: 1), dove α è una costante scelta a priori.

2.6 Scaling

Come visto nelle sezioni precedenti, molti modelli non sono invarianti per trasformazioni lineari delle singole features, quindi dipendono dall'unità di misura utilizzata. Dato che le funzioni di errore spesso si basano sulla distanza euclidea, se i valori di una features sono di diversi ordini di grandezza più grandi di quelli delle altre, gli errori verranno dominati da questa feature. Nel caso del coordinate descent (sezione 2.5.2), la mancanza di scaling può anche portare ad una instabilità numerica dell'algoritmo. Per mitigare questo problema sono stati sviluppati diversi metodi.

2.6.1 Standard scaling

Lo standard scaling è una trasformazione che ha come obbiettivo quello di portare tutte le features ad una media 0 e deviazione standard 1. Ogni feature $x_{k(i)}$ (con $k = 1, \cdots, p$ e $i = 1, \cdots, n$) viene quindi sostituita con $x'_{k(i)}$ definita da

$$x'_{k(i)} = \frac{x_{k(i)} - \sum_{j=1}^{n} x_{k(j)}}{\sqrt{\sum_{j=1}^{n} x^2_{k(j)}}}. \tag{2.18}$$

Possiamo vedere che in questo modo la media empirica di ogni feature si annulla e la deviazione standard di ogni feature diventa 1, portando tutte le misure allo stesso ordine di grandezza.

2.6.2 Min Max scaling

Un altro metodo di scaling è quello del min max scaling. Questo metodo consiste nel portare tutte le features nella finestra $[0, 1]$. La trasformazione è definita da

$$x'_{k(i)} = \frac{x_{k(i)} - \min x_k}{\max x_k - \min x_k}. \tag{2.19}$$

2.7 Riduzione di dimensionalità

2.7.1 PCA

Lo scopo della PCA è effettuare una trasformazione lineare ortonormale sul dataset $\mathbf{X}$ per ottenere un dataset $\mathbf{T}$ tale da avere sulla prima componente la varianza più grande possibile, e ogni altra componente deve avere la varianza più grande possibile con la restrizione che sia linearmente indipendente dalle componenti prima [9]. Consideriamo una matrice $\mathbf{X}$ con media 0 sulle righe e definiamo $\mathbf{T} = \mathbf{XW}$. Abbiamo quindi che la trasformazione mappa ogni vettore riga $\mathbf{x}_{(i)}$ di $\mathbf{X}$ ad un vettore $\mathbf{t}_{(i)} = (t_1, \cdots, t_l)_{(i)}$ definita da

$$t_{k(i)} = \mathbf{x}_{(i)} \cdot \mathbf{w}_k, \tag{2.20}$$

con la restrizione che $|\mathbf{w}_k| = 1$. Per trovare il primo vettore $\mathbf{w}_{(1)}$ imponiamo che

$$\begin{aligned}
\mathbf{w}_{(1)} &= \underset{|\mathbf{w}|=1}{arg\,max} \left\{ \sum_{i=1}^{n} t_{1(i)}^2 \right\} && (2.21)\\
&= \underset{|\mathbf{w}|=1}{arg\,max} \left\{ \sum_{i=1}^{n} \left(\mathbf{x}_{(i)} \cdot \mathbf{w} \right)^2 \right\} && (2.22)\\
&= \underset{|\mathbf{w}|=1}{arg\,max} \left\{ |\mathbf{Xw}|^2 \right\} && (2.23)\\
&= \underset{|\mathbf{w}|=1}{arg\,max} \left\{ \mathbf{w}^T \mathbf{X}^T \mathbf{Xw} \right\}. && (2.24)
\end{aligned}$$

Per risolvere questo problema di massimizzazione, procediamo utilizzando la tecnica dei moltiplicatori di Lagrange. Scriviamo la lagrangiana come

$$\mathscr{L} = \mathbf{w}^T \mathbf{X}^T \mathbf{Xw} - \lambda \left(\mathbf{w}^T \mathbf{w} - 1 \right), \tag{2.25}$$

e derivando rispetto a $\mathbf{w}$ otteniamo che

$$\mathbf{X}^T \mathbf{Xw} = \lambda \mathbf{w}, \tag{2.26}$$

dalla quale vediamo che $\mathbf{w}$ deve essere un autovettore di $\mathbf{X}^T\mathbf{X}$. La varianza di $\mathbf{t}_1$ sarà quindi $|\mathbf{Xw}|^2 = \lambda^2$, quindi per massimizzarla dobbiamo prendere l'autovettore corrispondente all'autovalore più grande. Per trovare il secondo vettore invece dobbiamo imporre che

$$\mathbf{w}_{(2)} = \underset{|\mathbf{w}|=1, \mathbf{w}_1 \cdot \mathbf{w}=0}{arg\,max} \left\{ \mathbf{w}^T \mathbf{X}^T \mathbf{Xw} \right\}, \tag{2.27}$$

per cui la lagrangiana sarà

$$\mathscr{L} = \mathbf{w}^T \mathbf{X}^T \mathbf{Xw} - \lambda_2 \left(\mathbf{w}^T \mathbf{w} - 1 \right) - \mu \left(\mathbf{w}_1^T \mathbf{w} \right). \tag{2.28}$$

Derivando rispetto a $\mathbf{w}$ troviamo che

$$\begin{aligned} \mathbf{X}^T\mathbf{X}\mathbf{w} &= \lambda_2\mathbf{w} + \mu\mathbf{w}_1 && (2.29) \\ 0 &= 0 + \mu, && (2.30) \end{aligned}$$

dove la seconda equazione è stata ottenuta moltiplicando a sinistra per $\mathbf{w}_{(1)}$ e imponendo l'ortogonalità tra i vettori $\mathbf{w}$ e $\mathbf{w}_{(1)}$. Vediamo dalla (2.29) che anche $\mathbf{w}_{(2)}$ deve essere un autovettore di $\mathbf{X}^T\mathbf{X}$. Dovendo essere ortogonale a $\mathbf{w}_{(1)}$ e dovendo massimizzare la varianza, l'autovettore da scegliere sarà quello corrispondente al secondo autovalore dal modulo più grande. Il procedimento per le altre componenti è analogo.

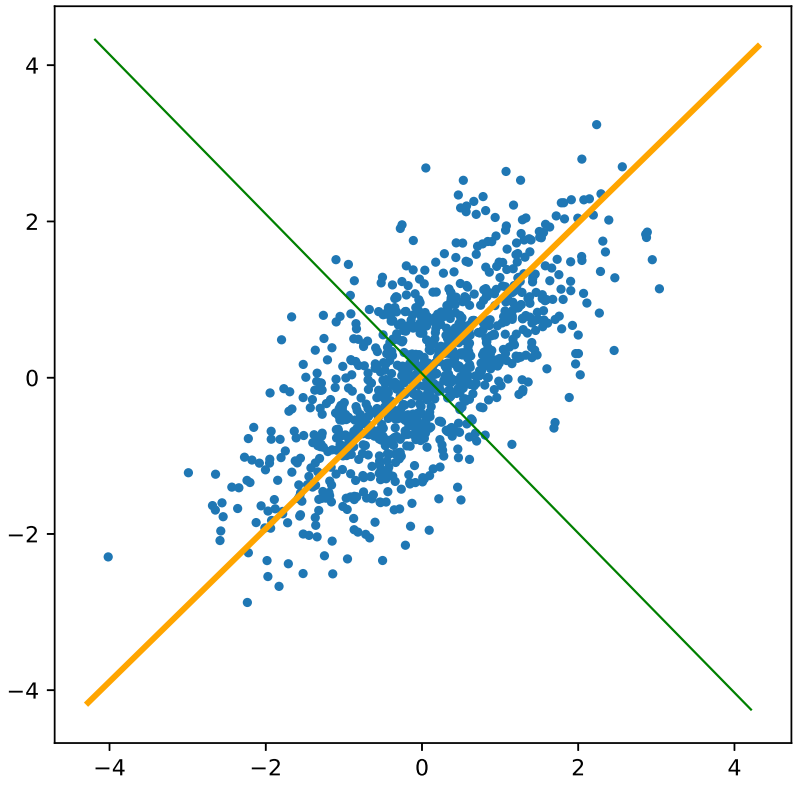

Figura 2.3: Esempio di PCA. L'asse di colore arancione è l'asse associato al primo valore principale, quello verde è associato al secondo valore principale.

2.8 Cross-validation

Introduciamo ora un metodo per stimare l'errore atteso introdotto nella sezione 2.4.1. Il Cross-Validation consiste nel dividere il set di training in K parti disgiunte di grandezza simile, come in fig. 2.4. Per ogni parte $k = 1, \cdots, K$, eseguiamo il fit del modello alle restanti $K - 1$ parti e lo valutiamo calcolando l'errore sul k-esimo set. In modo più formale, il procedimento è il seguente: sia $g : \{1, \cdots, n\} \to \{1, \cdots, K\}$ la funzione che associa ad ogni misura il suo indice k, sia inoltre $\hat{f}^{-k}(\mathbf{x})$ il risultato della regressione utilizzando tutti i dati tranne quelli con indice k. La stima dell'errore atteso è allora

$$\mathrm{CV} = \frac{1}{n} \sum_{i=1}^{K} \left[\frac{1}{\#g(i)^{-1}} \sum_{j \in g(i)^{-1}} \left(y_i - \hat{f}^{-k}(\mathbf{x}_i) \right) \right]. \tag{2.31}$$

Tipicamente $K = 5$ o $K = 10$ e viene comunque scelto dispari.

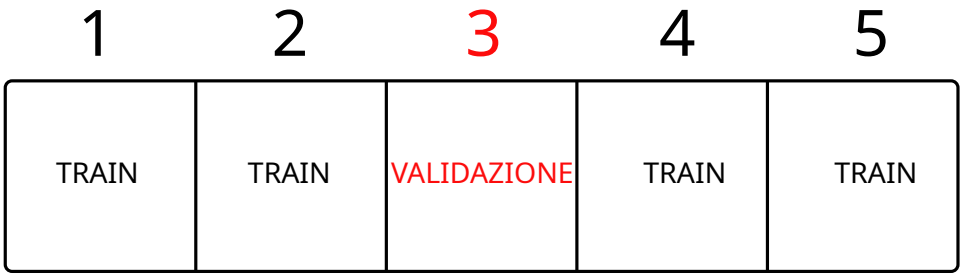

Figura 2.4: Nel cross-validation il set di training viene diviso in un numero K di parti disgiunte. In questa immagine possiamo vedere un esempio con $K = 5$.

2.9 Selezione degli iperparametri

I parametri di un modello che non vengono ottimizzati durante la fase di learning vengono chiamati iperparametri. Un esempio di iperparametro è il coefficiente λ della regressione ridge. Vediamo ora due metodi per la scelta di questi parametri.

2.9.1 Grid Search

Nella grid search, prima si scelgono un numero finito di valori ragionevoli per ciascun iperparametro, poi si provano tutte le combinazioni di iperparametri. Per ogni combinazione viene misurata una qualche metrica come l'errore quadratico medio utilizzando una cross-validation. Dopo aver trovato i migliori iperparametri, si può procedere ad eseguire il training sul set completo dei dati. Uno svantaggio della grid search è che il numero di training cresce rapidamente con il numero di iperparametri.

2.9.2 Random Search

Nella random search, invece di cominciare con una lista esaustiva di combinazioni di iperparametri, vengono definite delle distribuzioni su ciascun iperparametro. Vengono quindi estratte casualmente un certo numero di combinazioni che vengono valutate utilizzando il cross-validation con una qualche metrica. È particolarmente utile quando si ha un gran numero di iperparametri, non essendo possibile fare una ricerca esaustiva.

2.10 Indici di bontà del fit

2.10.1 r^2

Il coefficiente di determinazione (r^2) è una misura statistica di quanto bene i risultati di una regressione approssimano i dati reali. È definito dalla formula

$$r^2 = 1 - \frac{\sum_i (y_i - f(\mathbf{x}_i))^2}{\sum_i (y_i - \bar{y})^2}. \tag{2.32}$$

Dalla formula possiamo vedere che $r^2 \in (-\infty, 1]$, dove un valore vicino ad 1 indica che la regressione approssima perfettamente i dati, un valore di 0 si ha quando il risultato della regressione è l'iperpiano orizzontale e si ha un valore negativo quando il modello è peggio dell'iperpiano orizzontale, indicazione che il modello è sbagliato.

2.10.2 Pearson R

Il coefficiente di correlazione di Pearson è una misura della correlazione lineare tra due variabili aleatorie X e Y. È definito come

$$r_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y - \bar{y})^2}}. \tag{2.33}$$

Come possiamo vedere dalla definizione $-1 \leq r \leq 1$. L'interpretazione di questo parametro si può vedere dalla figura (2.5):

- se $r = -1$ le variabili x e y giaciono su una retta con coefficiente angolare negativo,
- se $r = 1$ le variabili x e y giaciono su una retta con coefficiente angolare positivo,
- se $r = 0$ le variabili x e y sono indipendenti.

Notiamo inoltre che la (2.33) è indipendente da una transazione e un riscalamento delle due variabili, cioè

$$r_{x,y} = r_{ax+b,cy+d}, \tag{2.34}$$

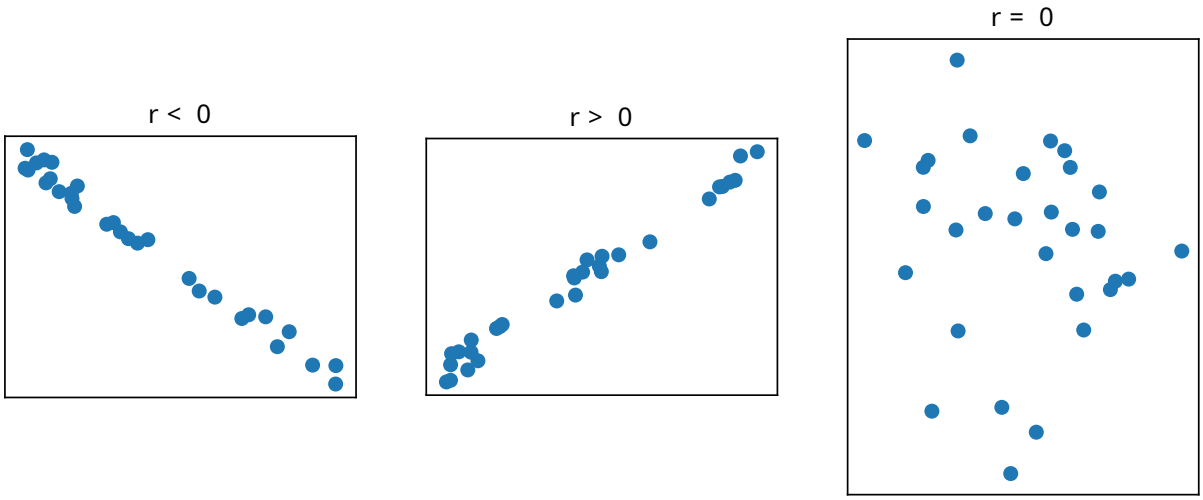

Figura 2.5: In questa immagine vengono mostrati tre differenti casi per r. Vediamo quando $r = 0$, le due variabili non sono correlate, quando $r > 0$ le variabili sono correlate e con $r < 0$ la correlazione tra le varibili è negativa

con a, b, c, d costanti e $a, c > 0$.

Ipotizzando che le x e le y sono estratte da due distribuzioni normali indipendenti si può dimostrare che la funzione densità di probabilità per il coefficiente di Pearson è [10]

$$p(r) = \frac{(1-r^2)^{n/2-2}}{B(1/2, n/2-1)}, \tag{2.35}$$

dove B è la funzione Beta ($B(x,y) = \int_0^1 t^{x-1}(1-t)^{y-1}dt$). Il grafico di questa funzione è mostrato in fig. 2.6. Possiamo quindi definire il p-value come la probabilità che il valore assoluto del coefficiente $|r'_{x',y'}|$ preso da due popolazioni x' e y' con correlazione 0 sia maggiore o uguale al valore misurato $|r_{x,y}|$. Un valore piccolo (< 0.05) del p-value indica che l'ipotesi nulla non spiega abbastanza bene le osservazioni.

2.11 Reti neurali

Le reti neurali sono un metodo relativamente recente nel campo del machine learning ma si sono dimostrate molto efficaci in svariati contesti. Una rete neurale è rappresentata da un grafo come quello in fig 2.7. Ogni nodo rappresenta un'unità computazionale che rappresenta una funzione semplice ed è chiamato neurone. Prendiamo in particolare un nodo come quello in fig 2.8: vediamo che un nodo ha N vertici in entrata, ciascuno con il suo peso e M vertici in uscita che saranno poi posti tutti allo stesso valore chiamato valore di attivazione. Chiamati x_i con $i = 1 \cdots N$ i valori sui vertici in uscita, il valore di

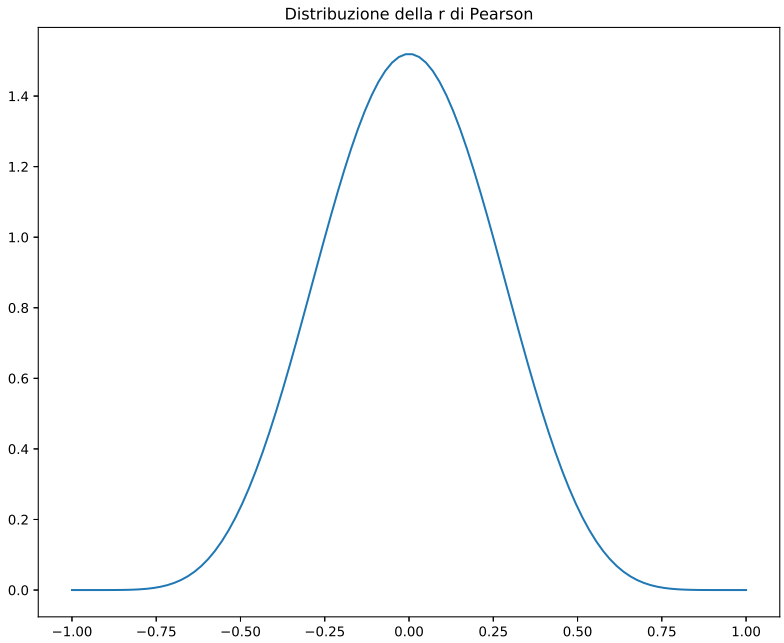

Figura 2.6: Distribuzione di probabilità della r di Pearson assumendo che x e y siano campionate da due distribuzioni normali indipendenti ($n = 17$).

attivazione è dato dalla formula

$$a = \text{act}\left(\sum_{i=1}^{N} w_i x_i + b\right), \tag{2.36}$$

dove act è una funzione differenziabile scelta a priori. Esistono diverse scelte per la funzione di attivazione, le più usate sono la funzione d'identità, la sigmoide definita come $\text{act}(x) = \frac{1}{1+\exp(-x)}$ e la ReLU, definita come $\text{act}(x) = \max(0, x)$. L'obbiettivo di una rete neurale è quindi quello di trovare i pesi w_i e i bias b di ciascun neurone che minimizza una qualche funzione di costo. Chiamando $\mathbf{W}^n_{ij}$ il peso che collega il j-esimo neurone del layer $n-1$ con il i-esimo neurone del layer n e $\mathbf{b}^n$ il vettore con i bias del layer n, vediamo che l'attivazione del layer n risulta

$$\mathbf{x}^n = \text{act}\left(\mathbf{W}^n \mathbf{x}^{n-1} + \mathbf{b}^n\right). \tag{2.37}$$

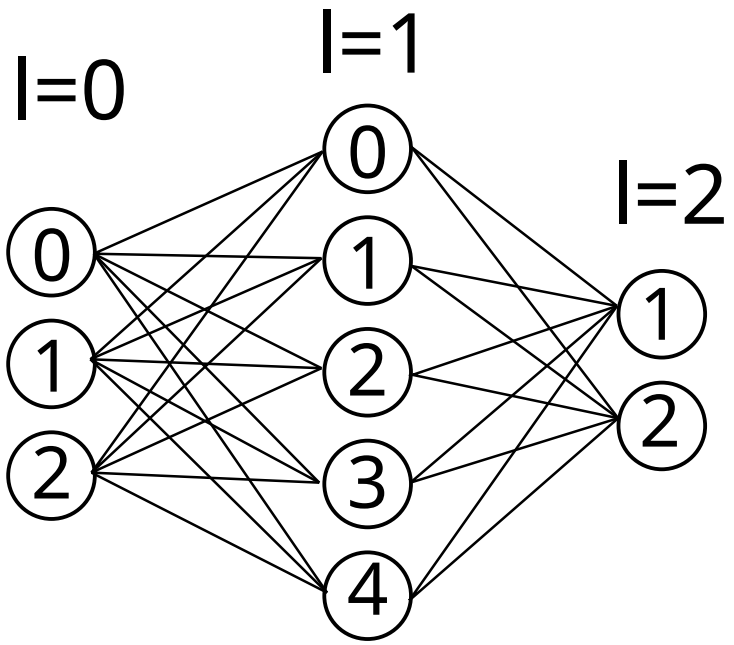

Figura 2.7: Esempio di possibile architettura per una rete neurale con 3 input e 2 output.

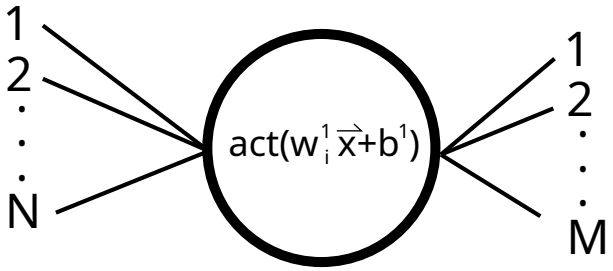

Figura 2.8: Funzionamento di un singolo neurone.

2.11.1 Backpropagation

Per utilizzare un metodo come il coordinate descent abbiamo bisogno della derivata parziale della funzione di costo rispetto ad ogni parametro della rete. Questa è ottenibile utilizzando la regola della catena: siano $\mathbf{y}(x_0, x_1, \cdots, x_n) : \mathbb{R}^n \to \mathbb{R}^m$ e $z(\mathbf{y}) : \mathbb{R}^m \to \mathbb{R}$, abbiamo allora che

$$\frac{\partial z}{\partial x_i} = \sum_{j=1}^{m} \frac{\partial z}{\partial y_j} \frac{\partial y_j}{\partial x_i}. \tag{2.38}$$

Per ottenere il gradiente della funzione di costo rispetto ai pesi della rete, si procede quindi in due passaggi:

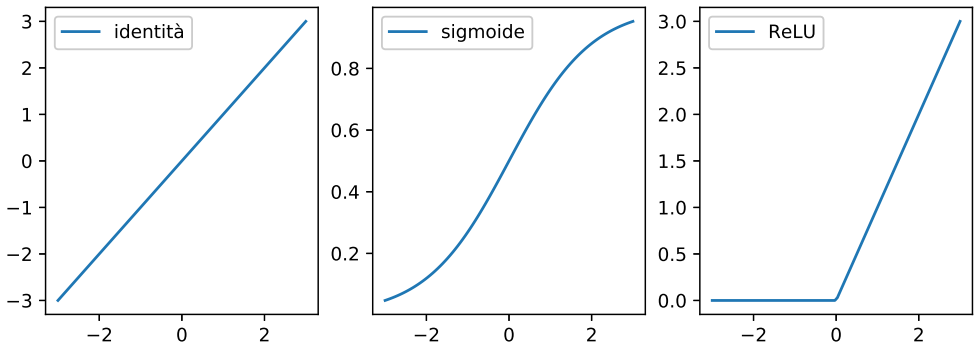

Figura 2.9: Esempi di diverse funzioni di attivazione.

1. durante il primo passaggio si calcola l'output della rete: questo passaggio è chiamato forward propagation;

2. durante il secondo passaggio si calcola il gradiente di ciascun peso utilizzando la regola della catena e i valori calcolati nel primo passaggio: questo passaggio si chiama backpropagation.

Normalmente il gradiente si calcola utilizzando tutti i dati nel set di training e viene poi utilizzato per aggiustare i pesi come nella tecnica del gradient descent. Nel campo delle reti neurali, ogni iterazione nella quale si calcola il gradiente viene chiamata *epoca*. Tutti questi passaggi sono implementati in diverse librerie, in particolare in questa tesi è stata utilizzata la libreria python TensorFlow.

2.11.2 Regolarizzazione

Anche le reti neurali, come le regressioni lineari, sono soggette al problema di overfitting. Un metodo per ridurlo è, come nella regressione ridge, di introdurre un termine di penalizzazione sul modulo quadrato dei pesi nella funzione di costo. Definiamo quindi una nuova funzione di costo:

$$C = C_0 + \frac{\lambda}{N} \sum_w w^2, \tag{2.39}$$

dove C_0 è la funzione di costo originale, λ è il parametro di regolarizzazione e la sommatoria è su tutti i pesi della rete. Altri metodi per ridurre l'overfitting sono:

- aumentare il dataset;

- diminuire la complessità della rete.

2.11.3 Regressione ridge come rete neurale

Notiamo che è possibile realizzare gli stessi risultati di una regressione ridge utilizzando una rete neurale senza layer tra l'input e l'output. Infatti, se prendiamo una rete come quella in fig. 2.10, utilizzando la funzione identità come attivazione, vediamo che l'output è

$$\text{out} = b + \sum_{i=1}^{N} x_i w_i. \tag{2.40}$$

Possiamo vedere che quest'ultima equazione è equivalente alla eq. 2.1. Vediamo inoltre che se utilizziamo una regolarizzazione sui pesi della rete, la funzione da minimizzare è l'eq. 2.39, che è equivalente alla eq. 2.11 della regressione ridge. Questo suggerisce che le prestazioni di una rete neurale con regolarizzazione siano le stesse o superiori di quelle di una regressione ridge.

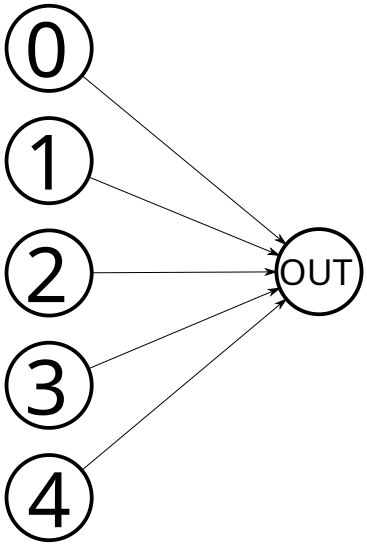

Figura 2.10: Rete per la realizzazione della regressione ridge.

2.11.4 Reti convoluzionali

Un problema che si incontra utilizzando una rete neurale con una topologia del tipo descritto nelle sezioni precedenti è quello della perdita delle informazioni di vicinanza tra le features. Infatti se le features rappresentano una serie temporale, con una topologia completamente connessa porta gli stessi risultati indipendentemente da come vengono mischiate le features in input. Inoltre dato l'elevato numero di parametri liberi in un layer completamente connesso la rete è più prona all'overfitting. Una soluzione a questi problemi sono i layer convoluzionali. Un layer convoluzionale ha i seguenti attributi:

- una serie di input in ingresso;
- una collezione di F vettori di dimensione K;

l'output di una rete convoluzionale si ottiene quindi applicando il kernel ad una finestra scorrevole sugli input: chiamando a_j il (j,f)-esimo output del layer convoluzionale e k_i^f l'i-esimo elemento del f-esimo kernel, abbiamo che

$$a_{jf} = \sum_{i=j}^{j+k} x_i k_i^f \tag{2.41}$$

per $j = 0, \ldots, n-k+1$. Abbiamo quindi un layer con KF parametri da ottimizzare. Normalmente in una rete neurale convoluzionale sono presenti più layer convoluzionali in successione, seguiti da uno o due layer completamente connessi.

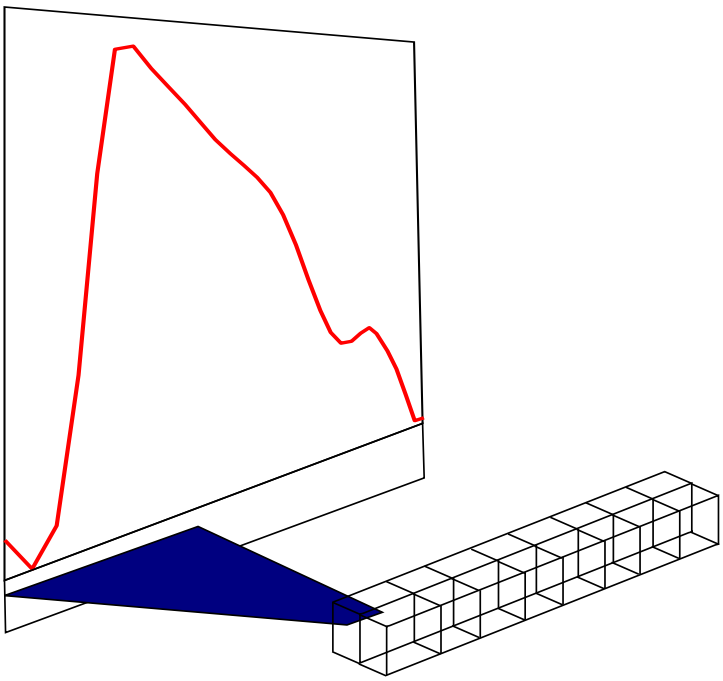

Figura 2.11: Immagine esplicativa di un layer convoluzionale.

Capitolo 3

Analisi dati

3.1 Dataset

I dati a disposizione consistono nel segnale misurato da una fotocamera di 4928 pazienti di età compresa tra i 18 e i 98 anni (media = 48.9, dev.std = 14.8) e vari metadati riguardanti i pazienti (sesso, fumatore, ...). Sono inoltre disponibili i dati dell'accelerometro dello smartphone al momento della presa dati.

3.2 Preprocessing

Per ogni paziente abbiamo un segnale RGB con valori nell'intervallo $[0, 1]$ per ciascuno dei 3 canali. Per estrarre il segnale PPG è stato utilizzato solamente il canale rosso, essendo quello che più si avvicina al colore dell'emoglobina. Un esempio di questo segnale è mostrato in fig. 3.1. Dalla figura del segnale possiamo notare diverse cose:

- il segnale è composto da un segnale periodico modulato da un segnale a bassa frequenza, causato sia da un contatto non buono con il fotosensore che dalla presenza di altri tessuti che da aggiustamenti automatici della luminosità eseguiti dal sensore;
- il segnale periodico non ha sempre la stessa ampiezza.

Per rimuovere la modulazione complessiva, è stata eseguita una media mobile con una finestra di 0.5 secondi ed è stata rimossa dal segnale originario, ottenendo un segnale come quello in fig. 3.2. Come si può notare, è ancora presente una modulazione moltiplicativa nel segnale, inoltre diversi segnali hanno ampiezze diverse. Per standardizzare l'ampiezza dei segnali, questi sono stati divisi per il loro inviluppo, ottenendo il segnale in fig. 3.3. Per trovare l'inviluppo è stato calcolato il valore assoluto $|H(t)|$ della trasformata di Hilbert del segnale, al quale è stato poi applicata una media mobile con una finestra di

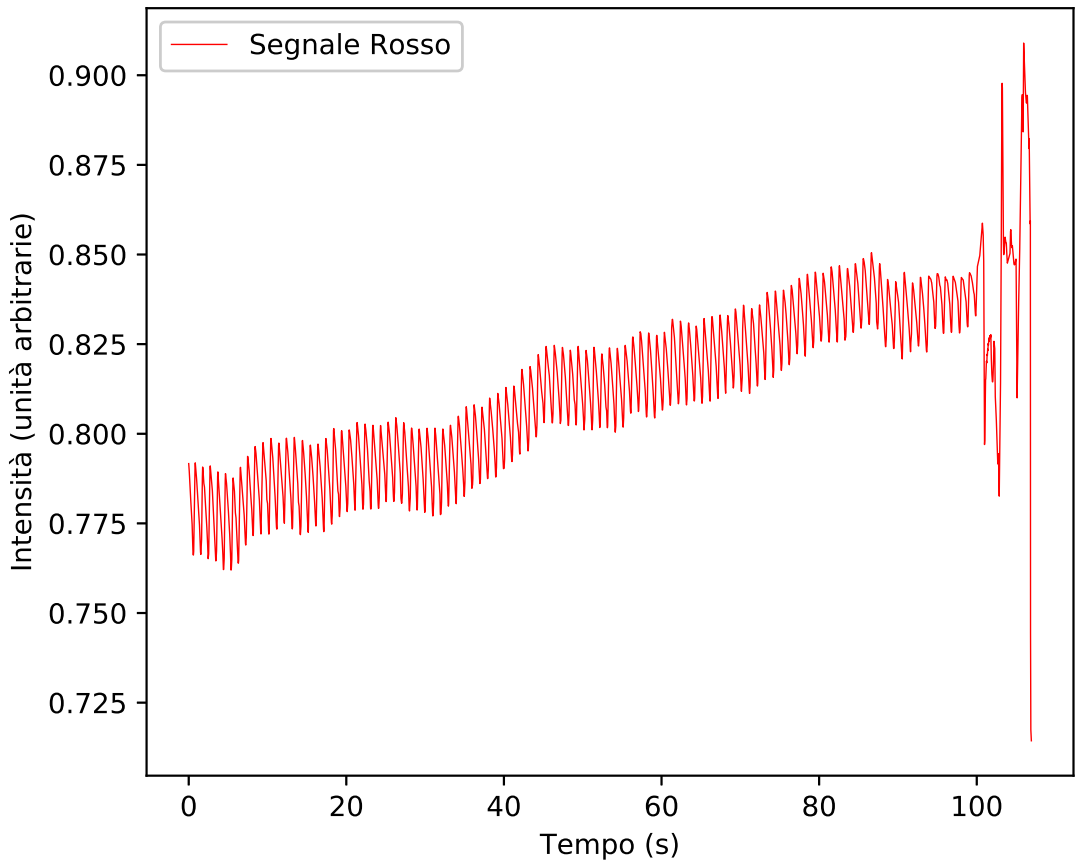

Figura 3.1: Segnale raw rilevato dal sensore di luminosità.

0.5 secondi. La trasformata di Hilbert per una funzione $f(\tau)$ è definita come

$$H(f)(t) = \frac{1}{\pi} \int_{-\infty}^{+\infty} \frac{f(\tau)}{t - \tau} d\tau. \tag{3.1}$$

3.3 Features extraction

Dal segnale filtrato nella sezione precedente sono stati individuati i picchi principali, corrispondenti al picco sistolico, utilizzando un algoritmo creato ad hoc. Una volta trovati i picchi, sono state estratte le distanze temporali tra due picchi consecutivi, chiamate distanze RR. Sono state calcolate anche le distanze tra due valori RR consecutivi, d'ora

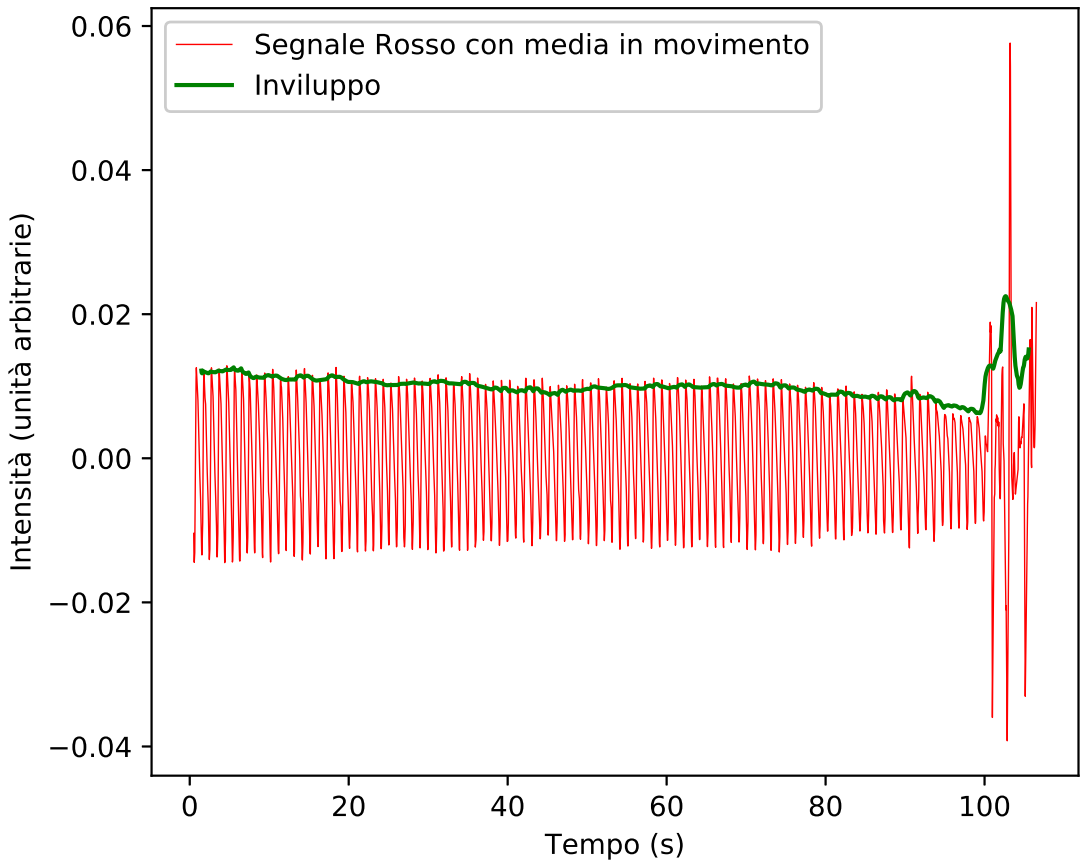

Figura 3.2: Segnale al quale è stata rimossa la media mobile per eliminare la parte a bassa frequenza. In verde è rappresentato l'inviluppo del segnale.

in poi chiamate RR_{diff}. Dal set di distanze RR, sono stati calcolati i seguenti indici statistici:

- media, chiamata in questo contesto Inter Beat Interval;
- mediana;
- deviazione standard;
- asimmetria;
- curtosi;

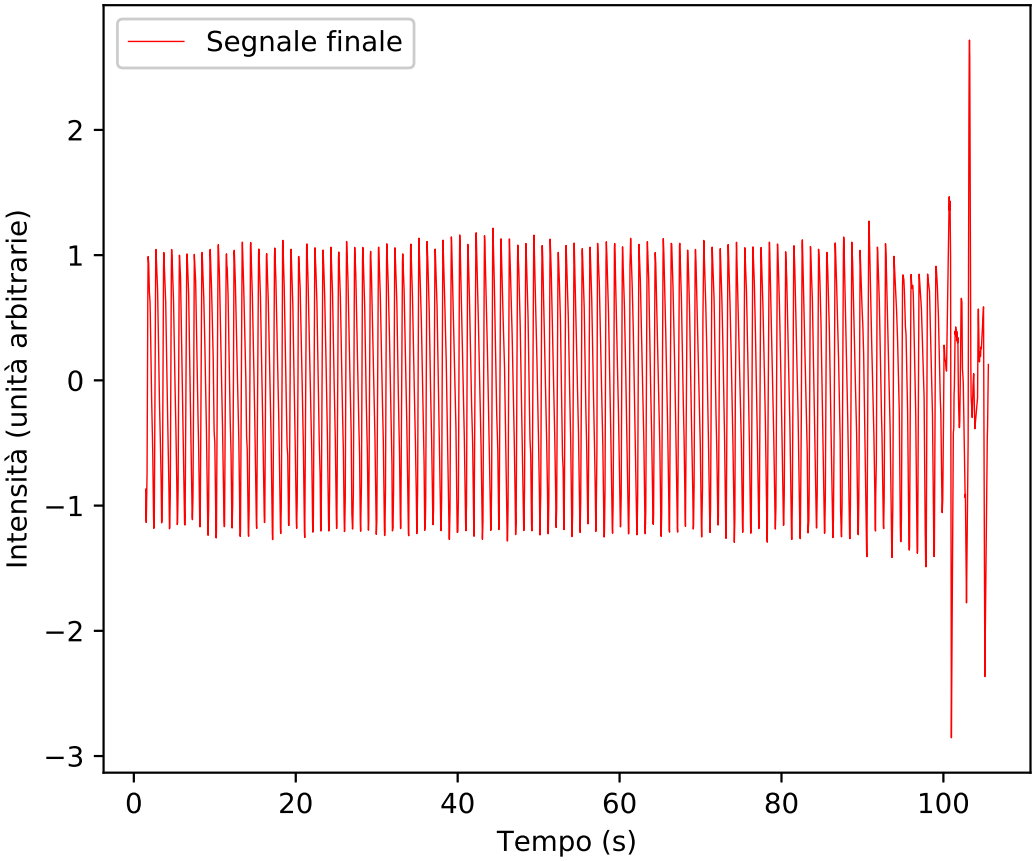

Figura 3.3: Segnale al termine del preprocessing.

- MAD (**M**edian **A**bsolute **D**eviation);
- TPR (**T**urning **P**oint **R**atio);
- Entropia di Shannon.

La MAD è definita come la mediana delle deviazioni dalla mediana del set:

$$\mathrm{MAD}(\{x_i\}) = \mathrm{mediana}\left(|x_i - \mathrm{mediana}(\{x_i\})|\right). \tag{3.2}$$

L'entropia di Shannon è stata calcolata come

$$H_{\mathrm{Shannon}} = -\sum_{i=0}^{N} \frac{RR_i}{\sum_{j=0}^{N} RR_j} \log_2 \left(\frac{RR_i}{\sum_{j=0}^{N} RR_j} \right) \tag{3.3}$$

Dal set RR_{diff} sono stati calcolati:

- TPR;
- PNN20, la percentuale di elementi in RR_{diff} che superano i $20ms$ in valore assoluto;
- PNN50, la percentuale di elementi in RR_{diff} che superano i $50ms$ in valore assoluto;
- $RMSSD$ (**R**oot **M**ean **S**quare **S**uccessive **D**ifferences);
- deviazione standard, chiamata in questo contesto $SDSD$ (**S**tandard **D**eviation **S**uccessive **D**ifferences).

La $RMSSD$ è stato calcolato utilizzando la formula

$$RMSSD = \sqrt{\left(\frac{1}{N}\sum_{i=0}^{N} RR_{\text{diff,i}}^2\right)} \tag{3.4}$$

Utilizzando la trasformata di fourier del segnale sono state calcolate anche le seguenti features:

HF l'integrale sulle frequenze comprese nell'intervallo $(0.15, 0.5]$;

LF l'integrale sulle frequenze comprese nell'intervallo $[0.04, 0.15]$.

3.3.1 SDPPG

Dalle features estratte dallo studio della derivata seconda sono stati calcolati:

- media e mediana dell'ampiezza del picco a;
- media e mediana dell'ampiezza del picco b;
- media e mediana dell'ampiezza del picco c;
- media e mediana dell'ampiezza del picco d;
- media e mediana dell'ampiezza del picco e;
- AGI, descritto nel paragrafo 1.5;
- mediana del rapporto tra l'altezza dei picchi b, c, d, e e il picco a;
- mediana dei valori $b - a$, $(b - e)/a$, $(c + d - b)/a$, $b - (d/a)$
- media e mediana delle differenze temporali tra:

  - picchi $a - b$;
  - picchi $b - c$;
  - picchi $c - d$;
  - picchi $d - e$;
  - picchi $a - c$;
  - picchi $a - d$;
  - picchi $a - e$;
  - picchi $b - d$;
  - picchi $b - e$;
  - picchi $c - e$;
- mediana delle pendenze delle rette che congiungono:
  - picchi $a - b$;
  - picchi $a - c$;
  - picchi $a - d$;
  - picchi $a - e$;
  - picchi $b - c$;
  - picchi $b - d$;
  - picchi $b - e$;
  - picchi $c - d$;
  - picchi $c - e$;
  - picchi $d - e$;

3.3.2 Metadati

Nel dataset sono inoltre presenti alcuni metadati. Quelli utilizzati sono:

- peso: il peso corporeo in kg;
- fumatore: 1 se l'individuo è un fumatore, -1 altrimenti;
- genere: 1 se l'individuo è maschio, -1 se femmina;
- afib: 1 se l'individuo è affetto da fibrillazione atriale, -1 altrimenti;

3.4 Pipeline

Per l'analisi dati sono state confrontate più pipeline. Nella prima sono state mantenute tutte le features, sono state riscalate utilizzando uno standard scaling (vedi 2.6.1) ed è stata poi effettuata una regressione ridge utilizzando una grid search sul parametro alpha (variandolo tra 0.001 e 10 con 1000 passi) e una cross-validation con 3 pieghe.

La seconda pipeline consiste in uno standard scaling, seguito da una PCA per ridurre la dimensionalità con un numero di componenti tali da avere almeno il 99% di varianza spiegata. Sono risultate quindi 52 features che sono state usate in una regressione ridge con una grid search impostata come nella prima pipeline.

Sono state poi testate alcune pipeline contenti reti neurali. La prima di queste utilizza uno standard scaling seguito da una rete neurale composto da due layer completamente connessi da 50 nodi ciascuno con la funzione identità come attivazione e un solo nodo di output.

La seconda utilizza uno standard scaling seguito da una PCA (99% di varianza spiegata) e la stessa rete neurale della terza pipeline. Le regressioni ridge, gli scaling e le PCA sono state eseguite utilizzando la libreria python *sklearn*, mentre le reti neurali sono state implementate in *Keras* e *TensorFlow*. Un riassunto delle pipeline utilizzate è in fig. 3.4.

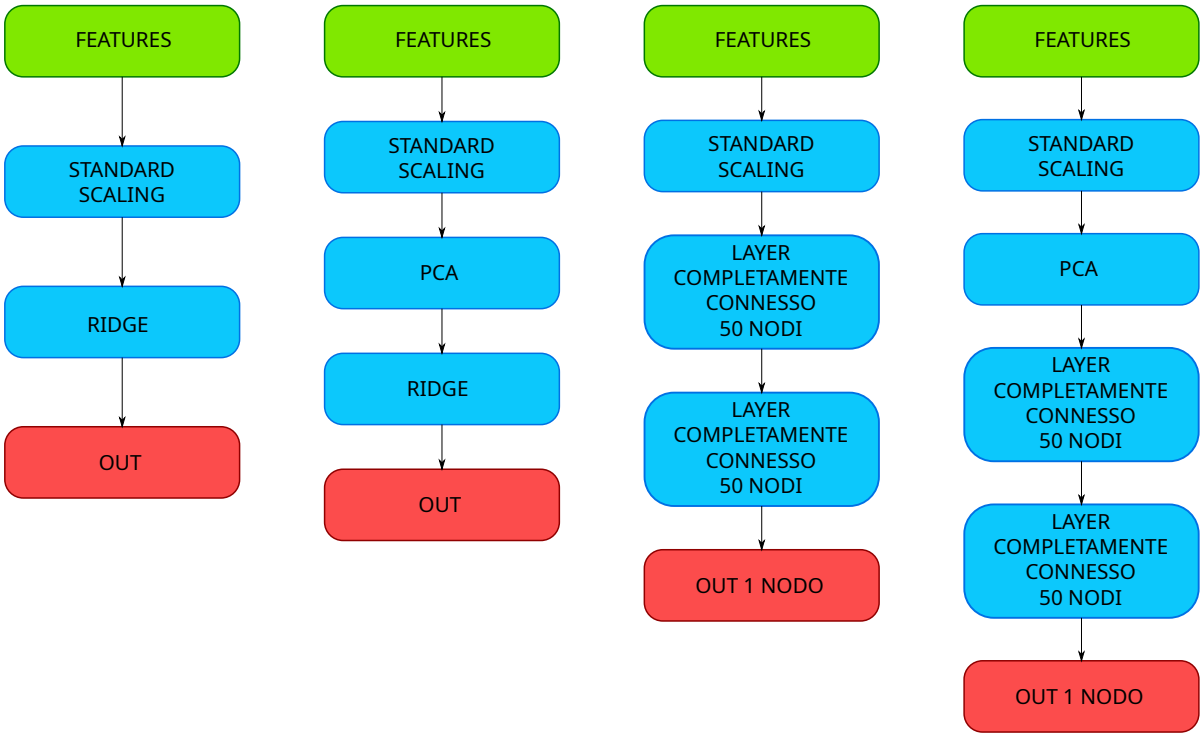

Figura 3.4: Immagine riassuntiva delle pipelines usate.

3.5 Risultati

I vari metodi illustrati nel paragrafo 3.4 sono stati valutati su un campione di testing contenente il 20% dell'intero dataset. Per le reti neurali sono state eseguite 120 epoche (vedi paragrafo 2.11.1). Come possiamo vedere dalla tabella 3.1, esiste una correlazione tra l'età predetta e l'età cronologica degli individui studiati, questo suggerisce che l'età del nostro algoritmo sia una buona indicazione dell'età biologica vera. Inoltre tutti i coefficienti di correlazione di Pearson hanno $p < 10^{-34}$, indicando una forte significatività statistica. Possiamo inoltre notare che la differenza di prestazione tra la regressione ridge e le reti neurali è minima, come già spiegato nel paragrafo 2.11.3. In fig. 3.8 sono rappresentati dei grafici con la relazione tra l'età reale in anni e quella predetta per ciascun individuo nel set di testing. La retta arancione rappresenta il fit lineare dei dati. Possiamo vedere come le età predette degli individui più giovani tendono a essere più alte di quelle cronologiche. Mentre il funzionamento delle reti neurali è praticamente una *black box*, per la regressione ridge senza PCA possiamo vedere quali features influenzano maggiormente il risultato finale. Essendo tutte le features normalizzate utilizzando uno standard scaling (paragrafo. 2.6.1), possiamo ordinare i pesi di ciascuna feature utilizzando come importanza il loro valore assoluto. In fig. 3.5 possiamo vedere un grafico a barre con le prime 10 features e i loro pesi. Possiamo notare che la prima feature è la mediana dell'AGI, un indice già studiato da [6] con buoni risultati. Possiamo anche notare che la sesta feature, la quale indica se l'individuo è fumatore o no, ha un peso negativo. Questo può essere spiegato dal fatto che nel dataset, i fumatori hanno un'età generalmente sotto la media (fig. 3.7).

	Pearson r		r^2	
	Train	Test	Train	Test
Ridge	0.7054	0.6600	0.4969	0.4347
Ridge PCA	0.6878	0.6884	0.4727	0.4675
NN	0.7068	0.6675	0.4996	0.4410
NN PCA	0.6814	0.6843	0.4643	0.4641

Tabella 3.1: Tabella riassuntiva con i risultati per ciascun metodo utilizzato.

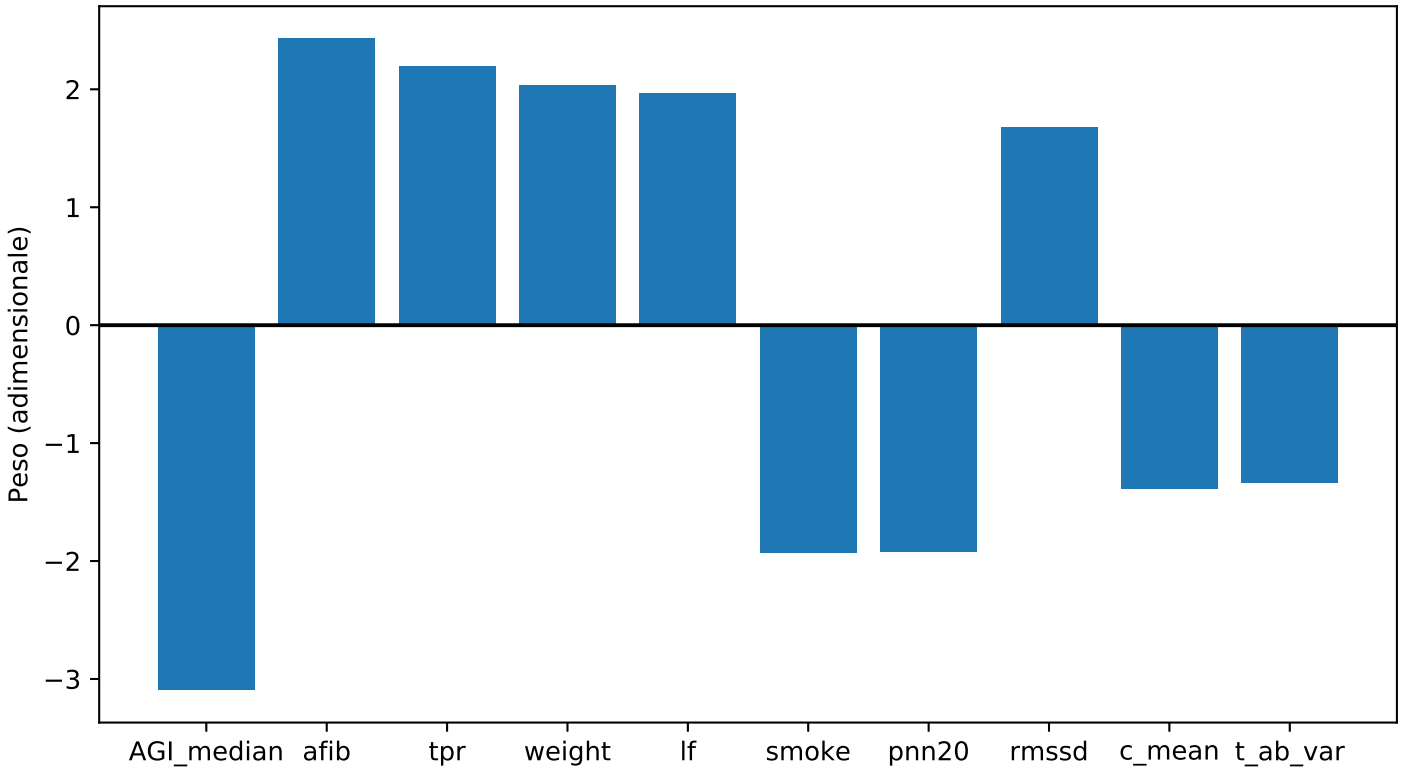

Figura 3.5: Grafico a barre con i pesi delle prime 10 features in ordine di valore assoluto.

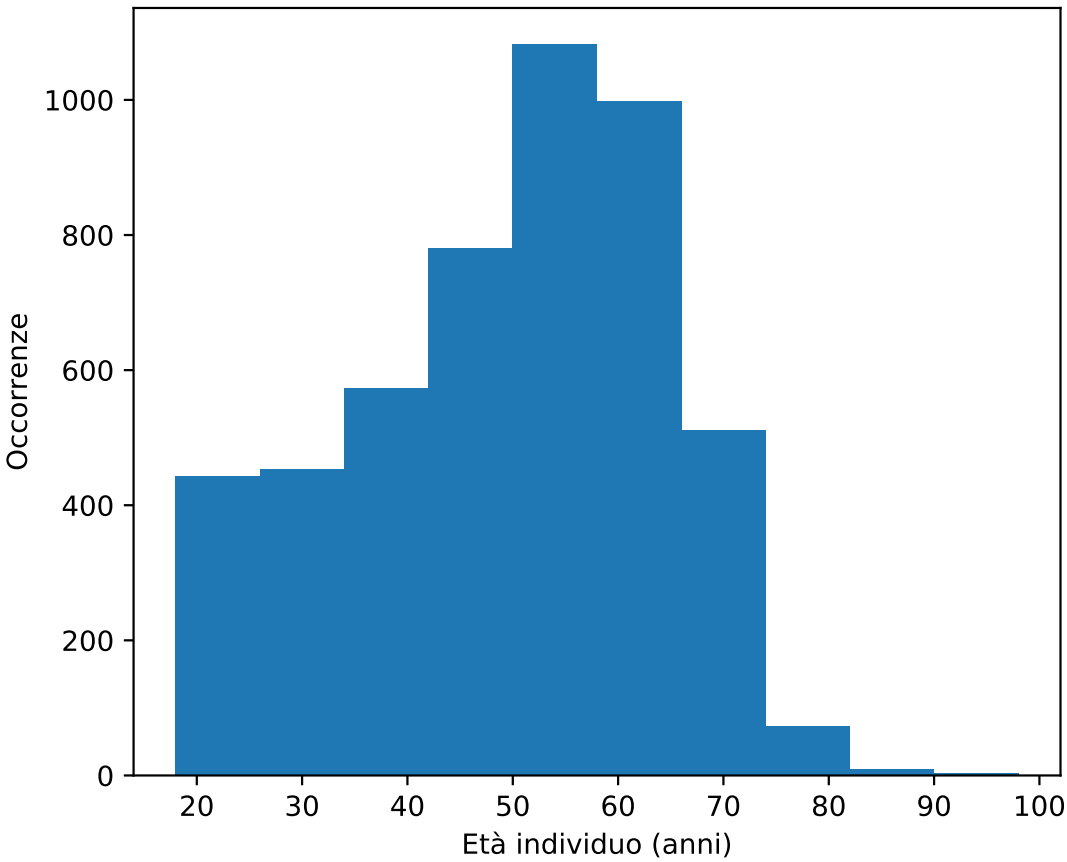

Figura 3.6: Distribuzione dell'età cronologica nel dataset.

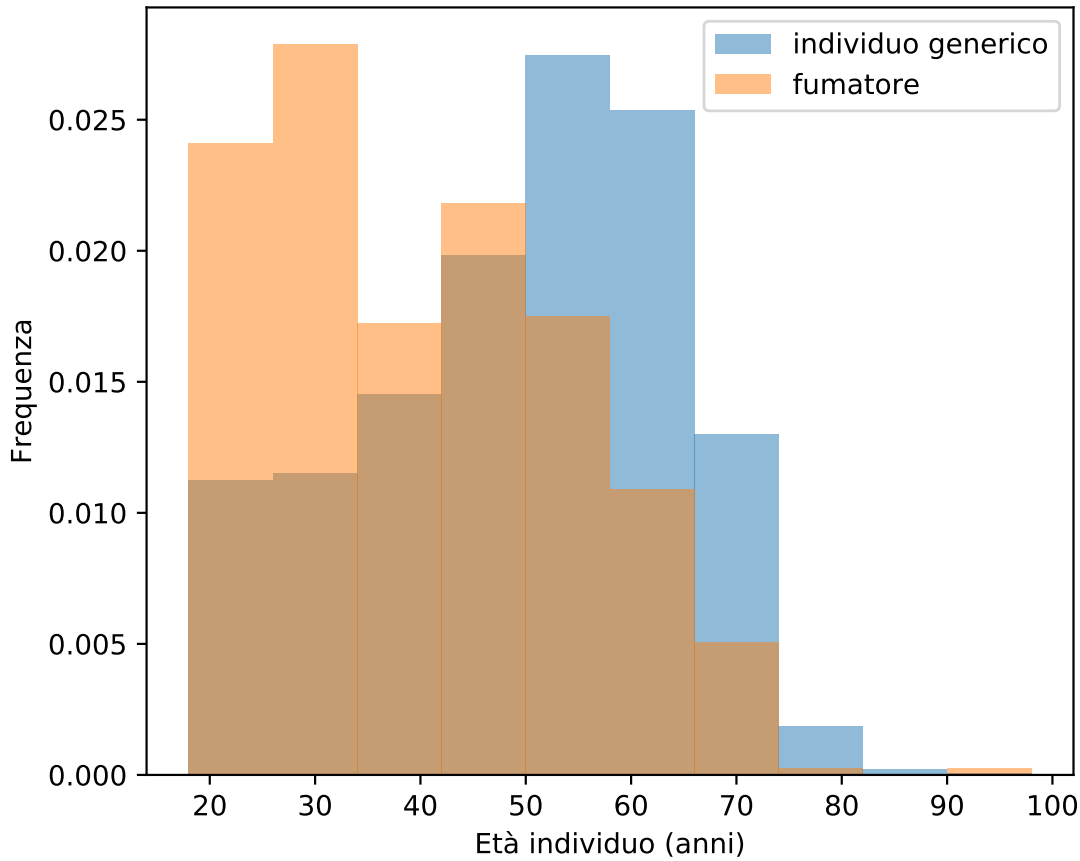

Figura 3.7: Istogramma normalizzato con l'età dei fumatori e complessiva. L'istogramma arancione rappresenta le frazioni di fumatori per ogni gruppo d'età. Si può notare come generalmente l'età dei fumatori è più bassa di quella dell'intero dataset.

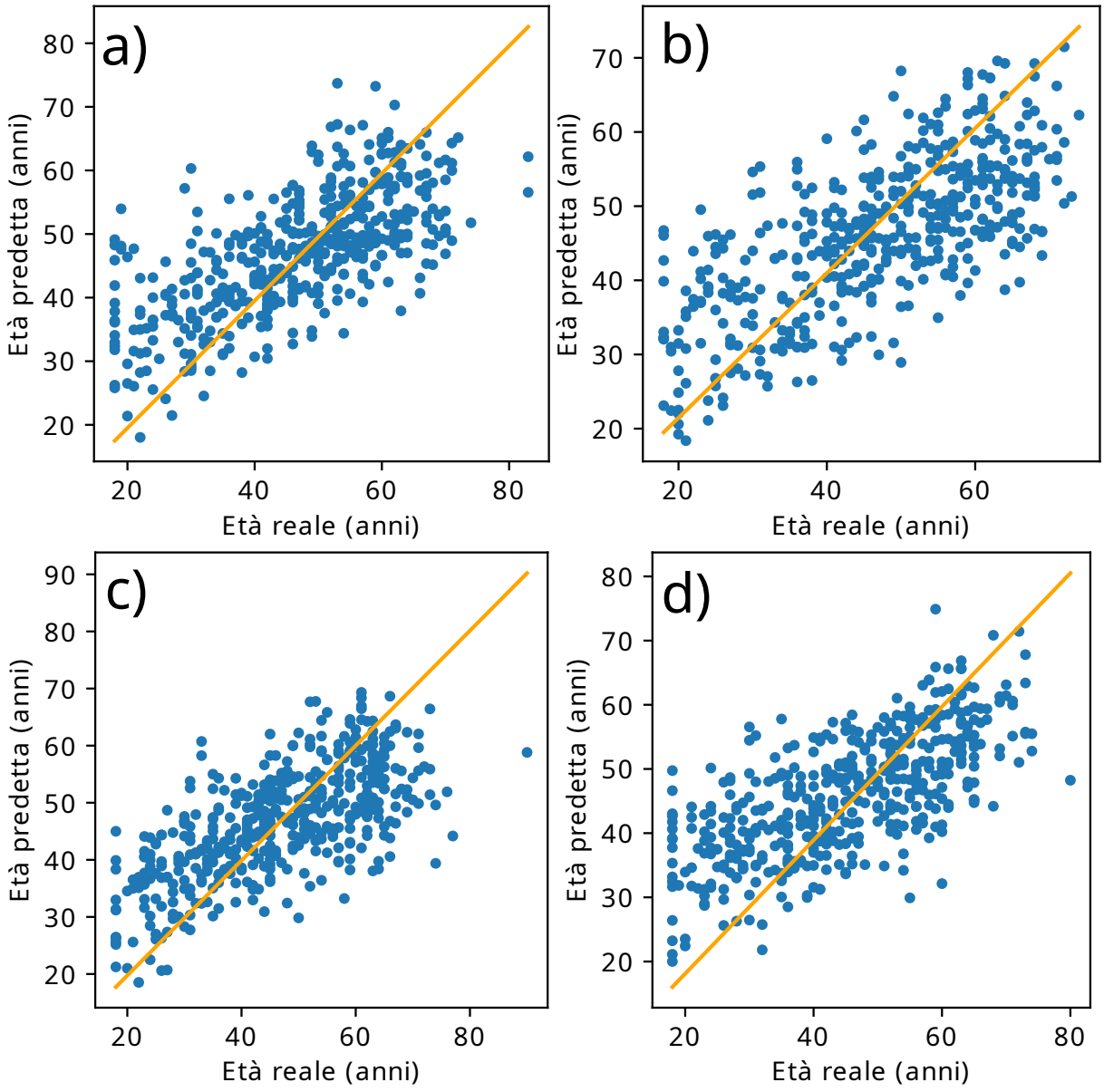

Figura 3.8: Diagramma con età predetta/età cronologica per ciascun individuo nel set di testing utilizzando a) la regressione ridge; b) una rete neurale; c) la regressione ridge con riduzione di dimensionalità; d) una rete neurale con riduzione di dimensionalità. La retta arancione è la retta di regressione.

Capitolo 4

Conclusioni

L'età predetta dagli algoritmi utilizzati mostra una buona correlazione con l'età cronologica. Questo suggerisce che le features utilizzate siano buoni indicatori utilizzabili per il calcolo dell'età biologica. Inoltre, analizzando i pesi della regressione ridge, siamo riusciti a confermare l'utilità delle features estratte dall'analisi della derivata seconda dei segnali di fotopletismografia, già studiati da [6]. Abbiamo inoltre sperimentato l'utilizzo di reti neurali con risultati paragonabili a quelli della regressione ridge. Essendo quest'ultima implementabile tramite reti neurali, esiste la possibilità che utilizzando una struttura diversa sia possibile raggiungere risultati migliori. I risultati ottenuti sono comunque di importanza medica poiché potranno essere utili come metodo economico e rapido per lo screening di possibili patologie o come indicatori di uno stile di vita non sano.

Bibliografia

[1] Bernard L Horecker. The absorption spectra of hemoglobin and its derivatives in the visible and near infra-red regions. *J. biol. Chem*, 148(1):173–183, 1943.

[2] Kirk Shelley, S Shelley, and Carol Lake. Pulse oximeter waveform: photoelectric plethysmography. *Clinical monitoring*, pages 420–428, 2001.

[3] AAR Kamal, JB Harness, G Irving, and AJ Mearns. Skin photoplethysmography—a review. *Computer methods and programs in biomedicine*, 28(4):257–269, 1989.

[4] Mohamed Elgendi. On the analysis of fingertip photoplethysmogram signals. *Current cardiology reviews*, 8(1):14–25, 2012.

[5] Nikos Stergiopulos, Berend E Westerhof, and Nico Westerhof. Total arterial inertance as the fourth element of the windkessel model. *American Journal of Physiology-Heart and Circulatory Physiology*, 276(1):H81–H88, 1999.

[6] Kenji Takazawa, Nobuhiro Tanaka, Masami Fujita, Osamu Matsuoka, Tokuyu Saiki, Masaru Aikawa, Sinobu Tamura, and Chiharu Ibukiyama. Assessment of vasoactive agents and vascular aging by the second derivative of photoplethysmogram waveform. *Hypertension*, 32(2):365–370, 1998.

[7] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[8] Stephen J Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

[9] Lindsay I Smith. A tutorial on principal components analysis. Technical report, 2002.

[10] Student. Probable error of a correlation coefficient. *Biometrika*, pages 302–310, 1908.