Analisi delle prestazioni dei computer

Mantovani Giacomo

28 novembre 2019

Indice

1 Introduzione

Lo scopo di questo esperimento è quello di prevedere le prestazioni di un computer in base alle sue componenti interne. A partire dal set di dati contenente informazioni relative alle componenti fisiche dei vari computer e le relative caratteristiche, traimite l'utilizzo di R, sono stati valutati due modelli di regressione, in particolare una regressione di tipo lineare e una non lineare, e successivamente effettuata una predizione del valore di uscita (nel nostro caso un valore che indica il livello complessivo delle prestazioni di un computer).

2 Dati

Il dati utilizzati nell'esperimento è stato reperito dal sito "UCI Machine Learning Repository" tramite il link:
https://archive.ics.uci.edu/ml/datasets/Computer+Hardware

2.1 Descrizione del set di dati

Il set di dati presenta al suo interno 10 colonne ed è composto da 209 record, i quali non contengono valori mancanti. In seguito è riportata una breve descrizione dei vari attributi della tabella:

- Vendor Name: Nome dell'azienda che ha prodotto la CPU
- Model Name: Valore unico per ogni record della tabella, rappresenta il modello della CPU.
- MYCT: Questo valore indica quanto tempo (in nanosecondi) impiega la CPU ad effettuare un ciclo di lettura/scrittura in memoria.
- MMIN: Indica la dimensione minima(in kilobytes) della memoria principale installabile sul computer.
- MMIN: Indica la dimensione massima (in kilobytes) della memoria principale installabile sul computer.
- CACH: Indica la dimensione della memoria CACHE in kilobytes.
- CHMIN: minimo numero di canali per unità
- CHMAX: massimo numero di canali per unità

- PRP: Published Relative Performance
- ERP: Estimated Relative Performance

2.2 Importazione e pulizia dei dati

Non è stata effettuata alcuna operazione sui dati precedente alla loro importazione su R. Prima di iniziare ad analizzarli abbiamo rimosso gli attributi che non verranno presi in considerazione durante l'esperimento, ossia "Vendor Name", "Model Name" ed "EPR". i primi due sono attributi che non influenzano le prestazioni del computer, mentre come valore per la predizione utilizzeremo l'attributo "PRP", di conseguenza il valore "EPR" non sarà utile ai fini dell'esperimento. Non è stata effettuata alcuna gestione dei valori mancanti in quanto non presenti nella tabella di dati. Inoltre la standardizzazione non è stata effettuata in quanto non modifica il grafico di dispersione.

```
> #Caricamento e pulizia del dataset.
> data = read.csv("compPerf.csv", sep =";")
> data$Vendor.Name<- NULL
> data$Model.Name<-NULL
> data$ERP <- NULL
>
```

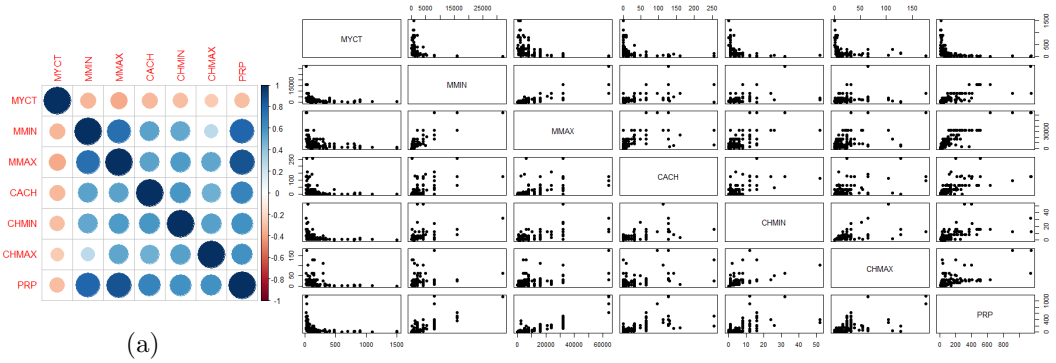
Figura 1: Codice R utilizzato per importare e pulire il set di dati.

	MYCT	MMIN	MMAX	CACH	CHMIN	CHMAX	PRP
1	125	256	6000	256	16	128	198
2	29	8000	32000	32	8	32	269
3	29	8000	32000	32	8	32	220
4	29	8000	32000	32	8	32	172
5	29	8000	16000	32	8	16	132
6	26	8000	32000	64	8	32	318

Figura 2: Prime righe del set di dati successive alla pulizia.

2.3 Visualizzazione dei dati

Il primo approccio esplorativo dei dati sarà di tipo grafico. Riportiamo di seguito il grafico relativo alla correlazione (a) e gli scatter plot tra i vari attributi (b).



(a) (b)

Figura 3: Correlazione tra gli attributi (a) e grafico di dispersione (b).

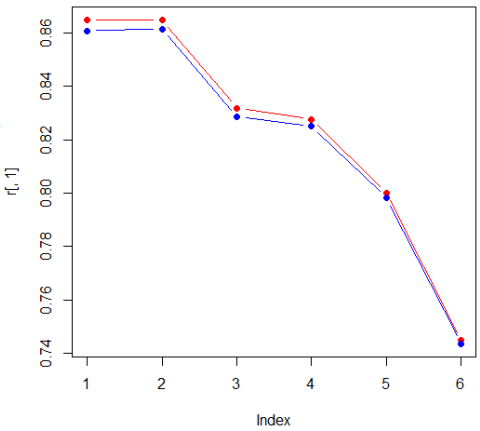
3 Analisi dei dati

3.1 Regressione lineare

Per applicare la regressione lineare partiamo dal modello originario e procediamo eliminando di volta in volta i fattori che presentano un p-value elevato. Ogni volta che un fattore viene eliminato dobbiamo controllare di quanto cala la varianza spiegata, nel caso avessimo eliminato un fattore importante per il modello considerato avremmo un calo significativo di quest'ultima.
Al termine di queste operazioni abbiamo raggiunto un valore di varianza spiegata pari a 0.80 utilizzando esclusivamente gli attributi MMAX e CACH. Abbiamo provato ad eliminare anche l'attributo CACH ma come possiamo notare dalla figura 3 (b) la varianza spiegata diminuisce di circa 0.06, di conseguenza è stato tenuto in considerazione nelle successive analisi.

```
> data.lm5 = lm(PRP~.-CHMIN-CHMAX-MYCT-MMIN,data = data)
> summary(data.lm5)

Call:
lm(formula = PRP ~ . - CHMIN - CHMAX - MYCT - MMIN, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-233.88  -31.44    6.22   30.45  420.42

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.756e+01  7.113e+00  -5.280 3.27e-07 ***
MMAX         9.776e-03  5.068e-04  19.291  < 2e-16 ***
CACH         1.105e+00  1.463e-01   7.555 1.35e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 72.25 on 206 degrees of freedom
Multiple R-squared:  0.8001,    Adjusted R-squared:  0.7982
F-statistic: 412.4 on 2 and 206 DF,  p-value: < 2.2e-16
```

(a)



(b)

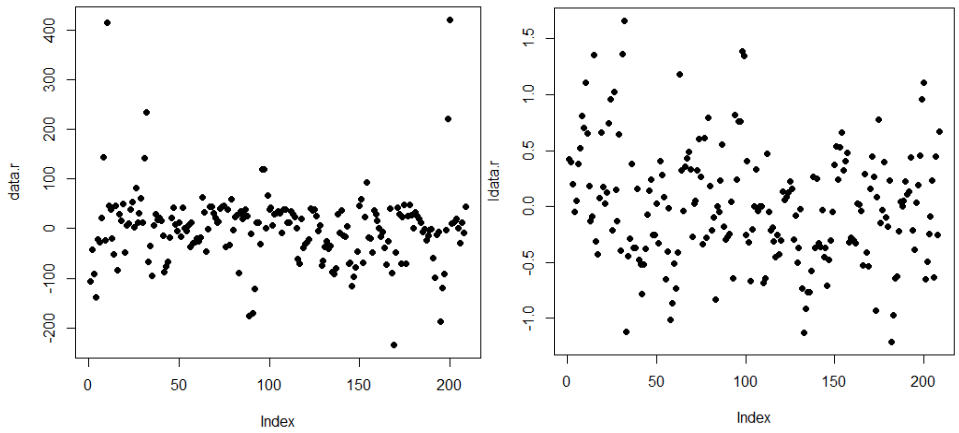
Figura 4: Eliminazione dell'ultimo attributo della regressione lineare (a) e variazione della varianza spiegata in seguito all'eliminazione dei fattori (b).

3.2 Regressione non lineare

Prima di procedere con l'analisi dei residui abbiamo verificato come i nostri dati si adattassero ad un modello di tipo non lineare seguendo gli stessi passaggi della regressione precedente ma applicando una funzione logaritmica al set di dati. In questo caso la varianza spiegata ottenuta con gli stessi due attributi (MMAX e CACH) risulta pari a 0.75.

3.3 Analisi dei residui

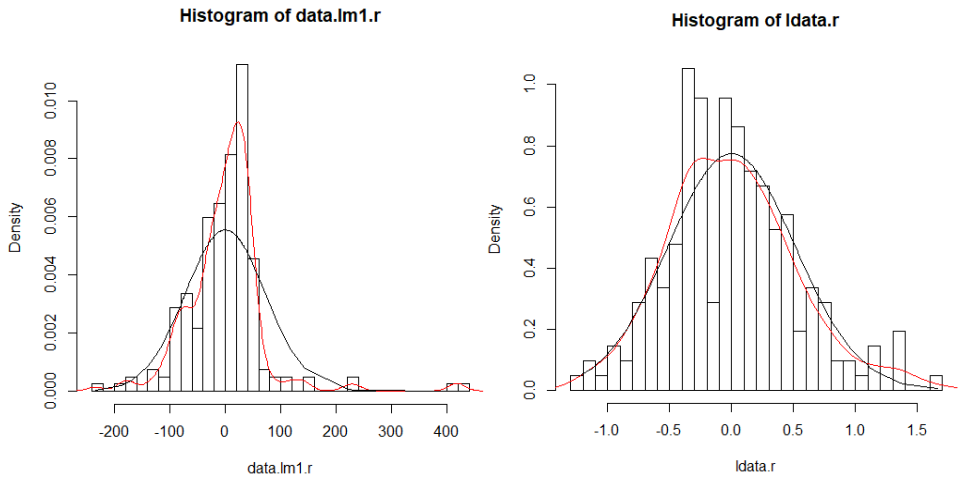
Dopo l'analisi di regressione eseguiamo alcuni test sui residui per avere una conferma di validità del modello, in particolare effettuiamo la stima della funzione di densità, calcoliamo la funzione cumulativa empirica, il test di Shapiro e Wilk ed il quantile-quantile plot.



(a) Regressione lineare (b) Regressione non lineare

Figura 5: Plot dei residui.

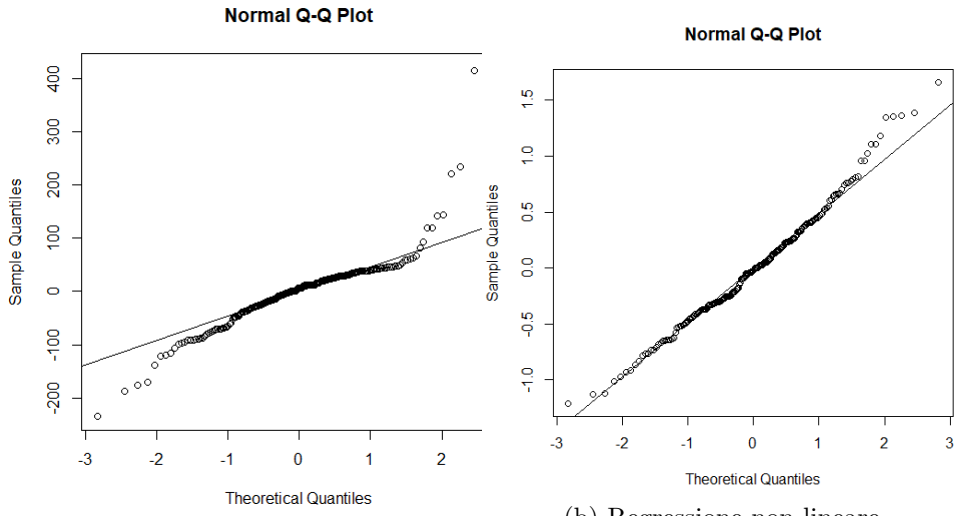
Andiamo a confrontare l'aderenza dei residui rispetto ad una distribuzione Gaussiana nel caso di regressione lineare (a) e non (b). Possiamo notare che nel caso di regressione non lineare abbiamo una distribuzione dei residui prossima alla distribuzione Gaussiana, ciò significa che la distribuzione dei residui è perlopiù casuale.



(a) Regressione lineare (b) Regressione non lineare

Figura 6: Osservazione della della densità empirica.

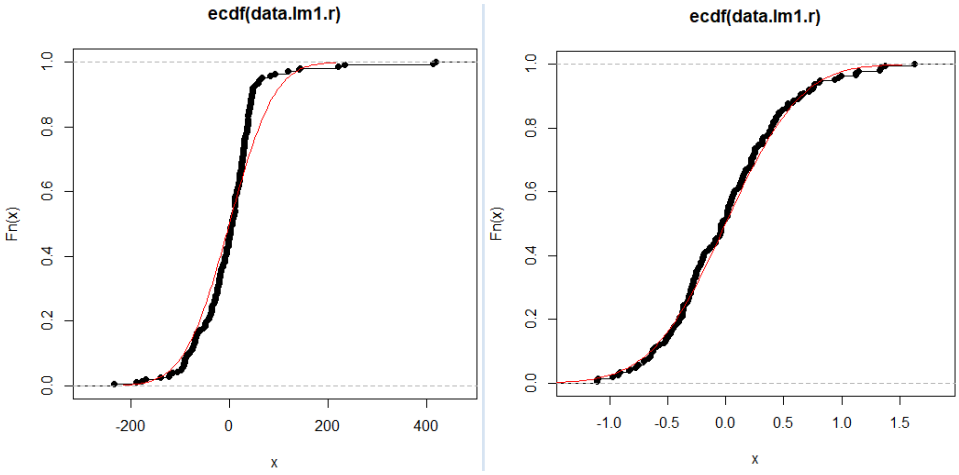
Q-Q plot: I punti si dispongono lungo una retta, ciò vuol dire che i dati si adattano bene al modello gaussiano.



(a) Regressione lineare

(b) Regressione non lineare

Figura 7: Osservazione della distribuzione dei quantili.



(a) Regressione lineare

(b) Regressione non lineare

Figura 8: Funzione cumulativa empirica.

Infine calcoliamo il test di Shapiro e Wilk ottenendo un valore pari a 0.83

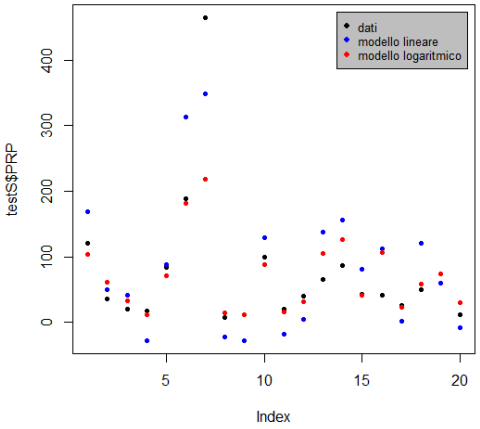
per il modello lineare e 0.98 per il modello non lineare. Essendo entrambi i valori prossimi ad 1, in particolare nel caso di regressione non lineare, risulta che i dati abbiano una distribuzione normale.
Dall'analisi dei residui possiamo dedurre che il modello non lineare approssima meglio il nostro set di dati nonostante la diminuzione della varianza spiegata. Procediamo con l'autovalutazione dei due modelli.

3.4 Autovalutazione e predizione

In questa fase andremo quindi a prelevare un numero di record casuali dal nostro set di dati (dal momento che non abbiamo nuovi dati da confrontare con il modello usiamo una parte del set di dati come test) per confrontare successivamente i valori predetti dal modello con i valori effettivi. Applichiamo quindi le regressioni sul nuovo set di dati, tentiamo di predire i valori dei record prelevati precedentemente e calcoliamo l'errore relativo, ottenendo un errore pari a 0.75 nel caso di regressione lineare e 0.12 nel caso di regressione non lineare, figura 9 (a).

```
> set.seed(4367)
> testset = sort(sample(209, 20))
>
> trainingSet <- data[-testset,]
> testS <- data[testset,]
>
> trainingSet.log <- ldata[-testset,]
> testS.log <- ldata[testset,]
>
> trainingSet.lm = lm(PRP~.-CHMIN-CHMAX-MYCT-MMIN,data = trainingSet)
> prediction <- predict (trainingSet.lm, testS)
> trainingSet.log.lm = lm(PRP~.-CHMIN-CHMAX-MYCT-MMIN,data = trainingSet.log)
> prediction.log <- predict(trainingSet.log.lm, testS.log)
> sqrt(mean((prediction - testS$PRP)^2)/mean(testS$PRP)^2)
[1] 0.7511621
> sqrt(mean((prediction.log - testS.log$PRP)^2)/mean(testS.log$PRP)^2)
[1] 0.1226785
```

(a)



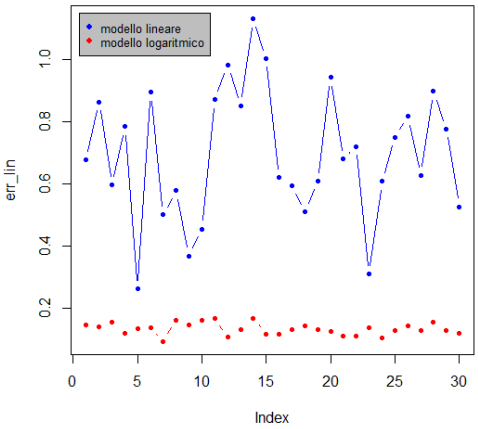
(b)

Figura 9: Codice R utilizzato per la predizione (a) e rappresentazione grafica dei valori predetti e valori effettivi (b).

Come ulteriore test andiamo ad effettuare lo stesso calcolo più volte e visualizziamo graficamente gli errori relativi ottenuti nei due modelli.

```
> n=30
>  err_lin = rep(0,n)
>  err_log = rep(0,n)
>  for(i in 1:n){
+  testset = sort(sample(209, 20))
+
+  trainingSet <- data[-testset,]
+  testS <- data[testset,]
+
+  trainingSet.log <- ldata[-testset,]
+  testS.log <- ldata[testset,]
+
+  trainingSet.lm = lm(PRP~.-CHMIN-CHMAX-MYCT-MMIN,data = trainingSet)
+  prediction <- predict (trainingSet.lm, testS)
+  trainingSet.log.lm = lm(PRP~.-CHMIN-CHMAX-MYCT-MMIN,data = trainingSet.log)
+  prediction.log <- predict(trainingSet.log.lm, testS.log)
+  err_lin[i] = sqrt(mean((prediction - testS$PRP)^2)/mean(testS$PRP)^2)
+  err_log[i] = sqrt(mean((prediction.log - testS.log$PRP)^2)/mean(testS.log$PRP)^2)
+ }
```

(a)



(b)

Figura 10: Codice R utilizzato per effettuare più test prelevando record casuali (a) e rappresentazione grafica degli errori relativi nel caso lineare e non (b).

4 Conclusioni

Dall'analisi effettuata risulta che il modello non lineare approssima meglio il set di dati utilizzato come si può notare dall'analisi dei residui (paragrafo 3.3). Inoltre abbiamo visto come l' errore relativo risulti minore nel caso di regressione non lineare. Possiamo quindi affermare che il modello non lineare ci permetta di ottenere previsioni dei dati più precise.