UNIVERSITÀ DELLA CALABRIA

DIPARTIMENTO DI INGEGNERIA INFORMATICA, MODELLISTICA, ELETTRONICA E SISTEMISTICA

DIMES

Corso di Metodi Statistici e Statistical Learning
Relazione Progetto Finale

Gruppo **F.C. Fuori Corso**

Francesco Curcio
Francesco Arci
Niccolò Bossio
Francesco Maria Cariello
Fabio Cusato
Manuel De Rose

A.A. 2023/2024

Indice

1 *Analisi dei dataset e obiettivi*

Il dataset su cui è stato svolto il progetto è stato scaricato dal sito: **https://archive.ics.uci.edu/ml/datasets/Gas+Turbine+CO+and+NOx+Emission+Data+Set**: esso contiene delle istanze di 11 misure provenienti dai sensori di una turbina a gas che si trova in Turchia, più precisamente nella regione nord-ovest di Marmara, allo scopo di studiarne la resa energetica.

Gli attributi presenti nel dataset sono i seguenti:

Attributo	**Abbreviazione**	**Unità di misura**
Temperatura dell'ambiente	AT	°C
Pressione dell'ambiente	AP	mbar
Umidità dell'ambiente	AH	%
Pressione differenziale del filtro dell'aria	AFDP	mbar
Contropressione della turbina a gas	GTEP	mbar
Temperatura d'ingresso della turbina	TIT	°C
Temperatura d'uscita della turbina	TAT	°C
Pressione di scarico del compressore	CDP	mbar
Resa energetica della turbina	TEY	MWh
Monossido di carbonio	CO	mg/m^3
Ossidi di nitrogeno	NOx	mg/m^3

1.1 Obiettivi

Gli obiettivi da portare a termine nel corso di questo elaborato sono i seguenti:

- Studiare quanto la resa energetica della turbina (TEY, Turbine Energy Yield) sia dipendente dalle caratteristiche ambientali. La variabile dipendente del modello di regressione risultante sarà TEY mentre gli altri attributi saranno i regressori.
- Analisi dell'eventuale multicollinearità tra i regressori presenti nei modelli costruiti sulla base dei dataset relativi agli anni 2011 e 2015;
- Confronto tra i modelli inferenziali ottenuti a partire dai dataset appena introdotti enfatizzando eventualmente differenze tra regressori e dati eventuali;
- Analisi relativa alla possibile presenza di eteroschedasticità;
- Costruzione di un modello predittivo che permetta di fornire le previsioni più accurate possibili per la variabile dipendente considerata.

1.2 Modelli inferenziali

Di seguito sono riportati i modelli stimati per i dataset relativi agli anni 2011 e 2015.

```
Call:
lm(formula = TEY ~ AT + AP + AH + AFDP + GTEP + TIT + TAT + CDP +
    CO + NOX, data = dataset)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.2979 -0.3759  0.0637  0.4366  2.6457 

Coefficients:
              Estimate Std. Error  t value Pr(>|t|)    
(Intercept) -1.332e+02  2.473e+00  -53.873  < 2e-16 ***
AT          -3.576e-01  2.823e-03 -126.650  < 2e-16 ***
AP          -6.814e-02  2.015e-03  -33.819  < 2e-16 ***
AH          -8.204e-03  9.035e-04   -9.080  < 2e-16 ***
AFDP        -4.857e-01  3.239e-02  -14.993  < 2e-16 ***
GTEP         3.235e-01  4.641e-02    6.970 3.43e-12 ***
TIT          6.196e-01  9.858e-03   62.854  < 2e-16 ***
TAT         -6.403e-01  1.566e-02  -40.901  < 2e-16 ***
CDP          1.324e+00  1.310e-01   10.107  < 2e-16 ***
CO           2.077e-02  7.321e-03    2.837  0.00457 ** 
NOX         -1.677e-02  1.372e-03  -12.219  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7285 on 7400 degrees of freedom
Multiple R-squared:  0.998,	Adjusted R-squared:  0.998 
F-statistic: 3.661e+05 on 10 and 7400 DF,  p-value: < 2.2e-16
```

Figura 1: Modello stimato per gt_2011

```
Call:
lm(formula = TEY ~ AT + AP + AH + AFDP + GTEP + TIT + TAT + CDP +
    CO + NOX, data = dataset)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.6594 -0.3333  0.0181  0.3531  2.4134 

Coefficients:
              Estimate Std. Error  t value Pr(>|t|)    
(Intercept) -1.831e+02  2.192e+00  -83.532  < 2e-16 ***
AT          -2.998e-01  2.294e-03 -130.682  < 2e-16 ***
AP          -5.815e-02  1.383e-03  -42.047  < 2e-16 ***
AH          -1.569e-02  7.816e-04  -20.072  < 2e-16 ***
AFDP        -8.549e-01  5.420e-02  -15.773  < 2e-16 ***
GTEP        -2.519e-02  5.361e-03   -4.698 2.67e-06 ***
TIT          6.809e-01  7.638e-03   89.147  < 2e-16 ***
TAT         -6.701e-01  1.118e-02  -59.959  < 2e-16 ***
CDP          1.583e+00  1.573e-01   10.062  < 2e-16 ***
CO           5.408e-02  6.438e-03    8.400  < 2e-16 ***
NOX         -2.678e-02  1.177e-03  -22.753  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.637 on 7373 degrees of freedom
Multiple R-squared:  0.9985,	Adjusted R-squared:  0.9984 
F-statistic: 4.755e+05 on 10 and 7373 DF,  p-value: < 2.2e-16
```

Figura 2: Modello stimato per gt_2015

2 *Analisi di multicollinearità per il dataset dell'anno 2011*

Un primo obiettivo riguarda l'analisi di multicollinearità nel modello di regressione lineare del dataset relativo all'anno 2011 (gt_2011). In questo contesto bisognerà valutare quindi la presenza di relazioni lineari quasi esatte tra le variabili esplicative, usando una serie di strumenti che permettono di verificare il rischio di multicollinearità nel modello.

2.1 *Analisi dei coefficienti di correlazione*

```
dataset <- read.csv("C:/Users/arcif/Desktop/Progetto Finale MS-SL/
    Datasets/gt_2011.csv", sep=",",header = TRUE)
cor(dataset)
```

	AT	AP	AH	AFDP	GTEP	TIT	TAT	TEY	CDP	CO	NOX
AT	1.00000000	-0.51688505	-0.55845855	-0.23734958	-0.03468239	0.02780909	0.22781013	-0.2042080	-0.09578242	-0.13791311	-0.65120752
AP	-0.51688505	1.00000000	0.05416229	0.21187711	0.11531169	0.05759195	-0.26360672	0.1974408	0.17085212	0.12218408	0.38986573
AH	-0.55845855	0.05416229	1.00000000	-0.09403554	-0.21002052	-0.22620869	0.07103691	-0.1153548	-0.18776385	0.13607242	0.20875590
AFDP	-0.23734958	0.21187711	-0.09403554	1.00000000	0.89063332	0.79000368	-0.74195598	0.9049759	0.89953479	-0.36395483	-0.03704827
GTEP	-0.03468239	0.11531169	-0.21002052	0.89063332	1.00000000	0.88343819	-0.80953841	0.9775101	0.99455598	-0.45563697	-0.24961712
TIT	0.02780909	0.05759195	-0.22620869	0.79000368	0.88343819	1.00000000	-0.45115895	0.9052405	0.89309999	-0.65009193	-0.23198437
TAT	0.22781013	-0.26360672	0.07103691	-0.74195598	-0.80953841	-0.45115895	1.00000000	-0.7675967	-0.80121541	0.04537919	0.08756568
TEY	-0.20420798	0.19744079	-0.11535481	0.90497591	0.97751013	0.90524047	-0.76759667	1.0000000	0.98844573	-0.47445896	-0.12640212
CDP	-0.09578242	0.17085212	-0.18776385	0.89953479	0.99455598	0.89309999	-0.80121541	0.9884457	1.00000000	-0.45705194	-0.19986218
CO	-0.13791311	0.12218408	0.13607242	-0.36395483	-0.45563697	-0.65009193	0.04537919	-0.4744590	-0.45705194	1.00000000	0.40805214
NOX	-0.65120752	0.38986573	0.20875590	-0.03704827	-0.24961712	-0.23198437	0.08756568	-0.1264021	-0.19986218	0.40805214	1.00000000

Dalla matrice di correlazione risaltano alcuni coefficienti, i quali si presentano con valori superiori a 0.8. In particolare si ha una forte correlazione tra i seguenti regressori:

- AFDP e GTEP (0.8906)
- CDP e AFDP (0.8906)
- TAT e GTEP (-0.8095)
- TIT e GTEP (0.8834)
- GTEP e CDP (0.9945)
- TIT e CDP (0.8930)
- CDP e TAT (-0.8012)

2.2 *Analisi del Determinante di X^tX*

```
y <- dataset$TEY
x1 <- dataset[,1]
x2 <- dataset[,2]
x3 <- dataset[,3]
x4 <- dataset[,4]
x5 <- dataset[,5]
x6 <- dataset[,6]
x7 <- dataset[,7]
x8 <- dataset[,9]
x9 <- dataset[,10]
x10 <- dataset[,11]

matrice <- cbind(rep(1,nrow(dataset)),x1,x2,x3,x4,x5,x6,x7,x8,x9,x10)
deter <- det(t(matrice) %*% matrice)
```

Il determinante della matrice X^tX può essere usato come indicatore di multicollinearità. In particolare, avere un valore del determinante prossimo allo zero ci porta a pensare che nel modello stimato vi sia multicollinearità. Nel nostro caso il determinante è pari a $5.110622e+48$.

2.3 Analisi del Condition Number

Il condition number è definito nel seguente modo:

$$k = \frac{\sqrt{max(\lambda_{\mathrm{h}})}}{\sqrt{min(\lambda_{\mathrm{h}})}}$$

Valori alti del condition number danno un'indicazione del rischio di multicollinearità. Infatti un valore alto di k può essere ottenuto a causa del minimo tra gli autovalori, prossimo allo zero. Empiricamente si osserva che un valore di k maggiore di 30 indica la presenza di multicollinearità.

```

autoval <-eigen(t(matrice)%*% matrice)
autoval$values
minAutoval <-min(autoval$values)
maxAutoval <- max(autoval$values)
conditionNumber <- sqrt(maxAutoval/minAutoval)
```

Nel nostro caso k è pari a 463397, perciò risulta essere maggiore di 30.

2.4 VIF e Tollerance

Per accertare la presenza di multicollinearità, esistono due strumenti analitici, denominati *Tollerance* e *VIF*:

$$Tollerance_j = 1 - R^2_{j,0}$$

$$VIF_j = \frac{1}{Tollerance} = \frac{1}{1 - R^2_{j,0}}$$

Inoltre, valori di $max(VIF_j) \geq 10$ segnalano la presenza di multicollinearità.
I risultati ottenuti sono i seguenti:

```
VIFModello <- vif(modelloStimato)
VIFModello
```

AT	AP	AH	AFDP	GTEP	TIT	TAT	CDP	CO	NOX
6.140467	2.244513	2.066524	6.417364	562.630282	353.229299	235.054163	314.860238	2.548134	3.000485

```
Tollerance <- 1/VIFModello
Tollerance
```

AT	AP	AH	AFDP	GTEP	TIT	TAT	CDP	CO	NOX
0.162854072	0.445530940	0.483904488	0.155827228	0.001777366	0.002831022	0.004254339	0.003176012	0.392444068	0.333279469

Si noti che il $max(VIF_j)$ è quello relativo a GTEP, ed è pari a 562.630282, risultando essere maggiore di 10.
A fronte dei test condotti è possibile concludere che il modello di regressione lineare presenta multicollinearità tra i valori osservati dei regressori.

2.5 Analisi del modello di regressione lineare

Un effetto della multicollinearità è quello di rendere alcuni regressori statisticamente non significativi, anche se in realtà tali regressori potrebbero essere significativi nello spiegare la variabile dipendente. In questa sezione verrà effettuata un'analisi dei test marginali e del test globale per quanto riguarda il modello di regressione lineare considerato. Di seguito è riportato il modello stimato:

```r
modelloStimato <- lm(TEY ~ AT+AP+AH+AFDP+GTEP+TIT+TAT+CDP+CO+NOX,dataset
  )
summary(modelloStimato)
```

```
Call:
lm(formula = TEY ~ AT + AP + AH + AFDP + GTEP + TIT + TAT + CDP +
    CO + NOX, data = dataset)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2979 -0.3759  0.0637  0.4366  2.6457

Coefficients:
              Estimate Std. Error  t value Pr(>|t|)
(Intercept) -1.332e+02  2.473e+00  -53.873  < 2e-16 ***
AT          -3.576e-01  2.823e-03 -126.650  < 2e-16 ***
AP          -6.814e-02  2.015e-03  -33.819  < 2e-16 ***
AH          -8.204e-03  9.035e-04   -9.080  < 2e-16 ***
AFDP        -4.857e-01  3.239e-02  -14.993  < 2e-16 ***
GTEP         3.235e-01  4.641e-02    6.970 3.43e-12 ***
TIT          6.196e-01  9.858e-03   62.854  < 2e-16 ***
TAT         -6.403e-01  1.566e-02  -40.901  < 2e-16 ***
CDP          1.324e+00  1.310e-01   10.107  < 2e-16 ***
CO           2.077e-02  7.321e-03    2.837  0.00457 **
NOX         -1.677e-02  1.372e-03  -12.219  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7285 on 7400 degrees of freedom
Multiple R-squared:  0.998,     Adjusted R-squared:  0.998
F-statistic: 3.661e+05 on 10 and 7400 DF,  p-value: < 2.2e-16
```

Dai risultati ottenuti si può notare che il test di adattamento complessivo restituisce un p-value minore di 0.05. Perciò esiste almeno un regressore statisticamente significativo nello spiegare la variabile dipendente. Osservando i test marginali si conclude che tutti i regressori risultano essere statisticamente significativi. Infine si deduce dall' R^2 che il 99.8% della variabilità totale del modello è spiegata dal modello stimato.
Concludiamo che, nonostante l'elevata multicollinearità del modello, essa non ha influito negativamente sul test.

3 *Analisi di multicollinearità per il dataset dell'anno 2015*

In questa sezione verrà eseguita un'analisi della multicollinearità relativa al dataset dell'anno 2015 (gt_2015), analoga a quella vista precedentemente.

3.1 Analisi dei coefficienti di correlazione

```
dataset <- read.csv("C:/Users/arcif/Desktop/Progetto Finale MS-SL/
   Datasets/gt_2015.csv", sep=",",header = TRUE)
cor(dataset)
```

	AT	AP	AH	AFDP	GTEP	TIT	TAT	TEY	CDP	CO	NOX
AT	1.0000000	-0.49309788	-0.46628847	0.46897582	0.19357758	0.33011162	0.20827660	0.10943666	0.20090892	-0.3906467	-0.59358018
AP	-0.4930979	1.00000000	0.08438144	-0.09414429	-0.04373034	-0.08160451	-0.29014662	0.05032596	0.02942009	0.2009446	0.21423632
AH	-0.4662885	0.08438144	1.00000000	-0.24545608	-0.29770831	-0.26068269	0.02625087	-0.18273177	-0.22170633	0.1589986	0.06535073
AFDP	0.4689758	-0.09414429	-0.24545608	1.00000000	0.84395757	0.91512777	-0.51980671	0.88495380	0.92299064	-0.6407886	-0.58445186
GTEP	0.1935776	-0.04373034	-0.29770831	0.84395757	1.00000000	0.89285131	-0.62065201	0.93233682	0.93814162	-0.5571773	-0.36665515
TIT	0.3301116	-0.08160451	-0.26068269	0.91512777	0.89285131	1.00000000	-0.39616119	0.95181259	0.95159003	-0.7380923	-0.52008080
TAT	0.2082766	-0.29014662	0.02625087	-0.51980671	-0.62065201	-0.39616119	1.00000000	-0.63393349	-0.65661298	0.0257682	0.05445541
TEY	0.1094367	0.05032596	-0.18273177	0.88495380	0.93233682	0.95181259	-0.63393349	1.00000000	0.99120733	-0.6167910	-0.40327795
CDP	0.2009089	0.02942009	-0.22170633	0.92299064	0.93814162	0.95159003	-0.65661298	0.99120733	1.00000000	-0.6126526	-0.44309256
CO	-0.3906467	0.20094462	0.15899855	-0.64078864	-0.55717729	-0.73809227	0.02576820	-0.61679097	-0.61265262	1.0000000	0.67839402
NOX	-0.5935802	0.21423632	0.06535073	-0.58445186	-0.36665515	-0.52008080	0.05445541	-0.40327795	-0.44309256	0.6783940	1.00000000

Anche in questo caso un primo step è quello di analizzare la matrice di correlazione, per vedere quali sono i regressori che presentano una dipendenza lineare quasi perfetta tra di loro. I coefficienti che superano valore 0.8 sono:

- AFDP e GTEP (0.8439)
- AFDP e TIT (0.9151)
- CDP e AFDP (0.9229)
- TIT e GTEP (0.8928)
- CDP e GTEP (0.9381)
- TIT e CDP (0.9515)

3.2 *Analisi del Determinante di* X^tX

```
y <- dataset$TEY
x1 <- dataset[,1]
x2 <- dataset[,2]
x3 <- dataset[,3]
x4 <- dataset[,4]
x5 <- dataset[,5]
x6 <- dataset[,6]
x7 <- dataset[,7]
x8 <- dataset[,9]
x9 <- dataset[,10]
x10 <- dataset[,11]

matrice <- cbind(rep(1,nrow(dataset)),x1,x2,x3,x4,x5,x6,x7,x8,x9,x10)
deter <- det(t(matrice) %*% matrice)
```

Il valore ottenuto del determinante è pari a $4.756515e + 49$. Essendo il suo valore molto lontano dallo zero, potrebbe essere un indicatore non affidabile della presenza di multicollinearità, perciò si prosegue con l'analisi di altre misure.

3.3 Analisi del Condition Number

Proseguiamo con il calcolo del condition number:

```
autoval <-eigen(t(matrice)%*% matrice)
autoval$values
minAutoval <-min(autoval$values)
maxAutoval <- max(autoval$values)
conditionNumber <- sqrt(maxAutoval/minAutoval)
conditionNumber
```

Il valore di k, pari a 467755.3, da un'indicazione del rischio di multicollinearità, in quanto risulta essere maggiore di 30.

3.4 VIF e Tollerance

```
VIFModello <- vif(modelloStimato)
VIFModello
Tollerance <- 1/VIFModello
Tollerance
```

AT	AP	AH	AFDP	GTEP	TIT	TAT	CDP	CO	NOX
6.273450	1.654196	2.037720	19.900118	10.466725	414.476920	68.454954	581.428160	3.766694	3.123017

Il VIF di valore massimo risulta essere quello associato al regressore CDP ed è pari a 581.42. Si tratta di un valore maggiore di 10, che dunque segnala il rischio di multicollinearità nel modello completo. Essendo la tollerance l'inversa del VIF è chiaramente osservabile che il valore più

AT	AP	AH	AFDP	GTEP	TIT	TAT	CDP	CO	NOX
0.159401927	0.604523445	0.490744588	0.050250959	0.095540866	0.002412680	0.014608147	0.001719903	0.265484774	0.320203146

prossimo allo zero è quello relativo al regressore CDP.

3.5 Analisi del modello di regressione lineare

Di seguito si riporta il modello completo, con tutti i regressori:

```
modelloStimato <- lm(TEY ~ AT+AP+AH+AFDP+GTEP+TIT+TAT+CDP+CO+NOX,dataset
    )
summary(modelloStimato)
```

```
Call:
lm(formula = TEY ~ AT + AP + AH + AFDP + GTEP + TIT + TAT + CDP +
    CO + NOX, data = dataset)

Residuals:
    Min      1Q  Median      3Q     Max
-3.6594 -0.3333  0.0181  0.3531  2.4134

Coefficients:
              Estimate Std. Error  t value Pr(>|t|)
(Intercept) -1.831e+02  2.192e+00  -83.532  < 2e-16 ***
AT          -2.998e-01  2.294e-03 -130.682  < 2e-16 ***
AP          -5.815e-02  1.383e-03  -42.047  < 2e-16 ***
AH          -1.569e-02  7.816e-04  -20.072  < 2e-16 ***
AFDP        -8.549e-01  5.420e-02  -15.773  < 2e-16 ***
GTEP        -2.519e-02  5.361e-03   -4.698 2.67e-06 ***
TIT          6.809e-01  7.638e-03   89.147  < 2e-16 ***
TAT         -6.701e-01  1.118e-02  -59.959  < 2e-16 ***
CDP          1.583e+00  1.573e-01   10.062  < 2e-16 ***
CO           5.408e-02  6.438e-03    8.400  < 2e-16 ***
NOX         -2.678e-02  1.177e-03  -22.753  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.637 on 7373 degrees of freedom
Multiple R-squared:  0.9985,    Adjusted R-squared:  0.9984
F-statistic: 4.755e+05 on 10 and 7373 DF,  p-value: < 2.2e-16
```

Le conclusioni raggiunte per tale modello sono analoghe a quelle viste precedentemente.Infatti, il test di adattamento complessivo restituisce un p-value minore di 0.05, perciò è presente almeno un regressore statisticamente significativo nello spiegare la variabile dipendente. Dai test marginali si conclude che tutti i regressori risultano essere statisticamente significativi e il valore dell' R^2, pari a 99.85%, indica che quasi tutta la variabilità del modello è spiegata dal modello stimato. Anche in questo caso si conclude che, nonostante l'elevata multicollinearità del modello, essa non ha influito negativamente sul test.

3.6 Confronto tra i due modelli

In entrambi i modelli si è visto che quello migliore è formato da tutti i regressori. Nel modello relativo al dataset 2011 si può osservare dalla matrice di correlazione che i regressori TAT e GTEP risultano altamente correlati e lo stesso vale per i regressori CDP e TAT, cosa che non si verifica nella matrice di correlazione del dataset relativo all'anno 2015. Inoltre è utile osservare il comportamento medio di alcuni regressori e della variabile dipendente dei due dataset.
Infatti, osservando i valori medi relativi ai regressori AT (temperatura), AP (pressione) e AH (umidità), in entrambi gli anni tali valori risultano pressochè identici. Ciò che differisce è il valore medio del rendimento energetico. Nell'anno 2011 troviamo un rendimento energetico medio pari a 135.8 MWh, mentre nel 2015 si ha un rendimento energetico medio di 133.9, evidenziando un decremento energetico medio dello 0.2%.

4 *Analisi di Eteroschedasticità*

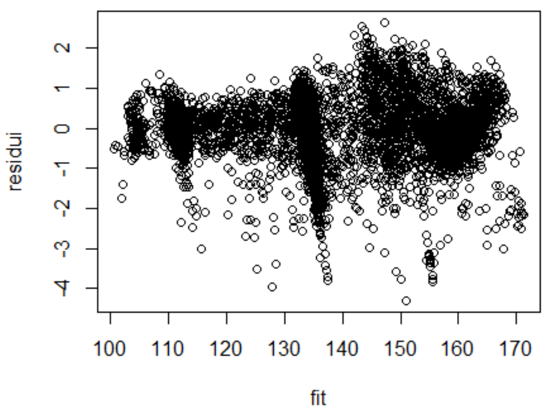
In questa sezione verrà affrontata l'analisi dell'eteroschedasticità del modello stimato relativo al dataset del 2011. Per analisi di eteroschedasticità si intende quel processo il cui obiettivo finale è scoprire se l'ipotesi fondamentale di errori omoschedastici all'interno del modello è violata. In questo contesto, poichè la varianza della variabile dipendente è uguale alla varianza dell'errore ϵ, se la prima dipende dai regressori anche la seconda dipenderà dai regressori. Quando ciò accade si parla di condizione di eteroschedasticità. L'analisi può essere condotta tramite la simbiosi di due strategie:

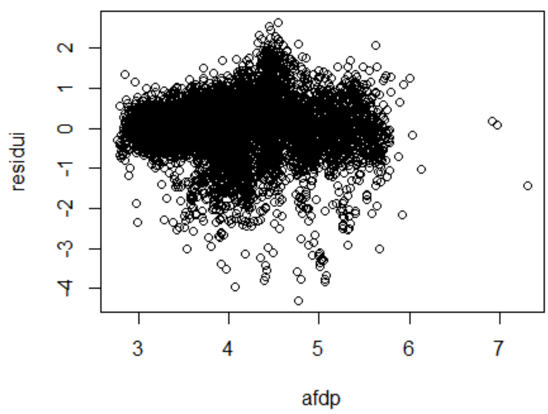
1. **Interpretazione qualitativa del grafico dei punti di coordinate** $(\hat{y_i}, \hat{\epsilon_i})$: in una situazione di eteroschedasticità ci si aspetta una qualche correlazione tra le ordinate stimate e le stime dei residui, e che quindi i punti nel grafico a dispersione non siano distribuiti in maniera casuale.

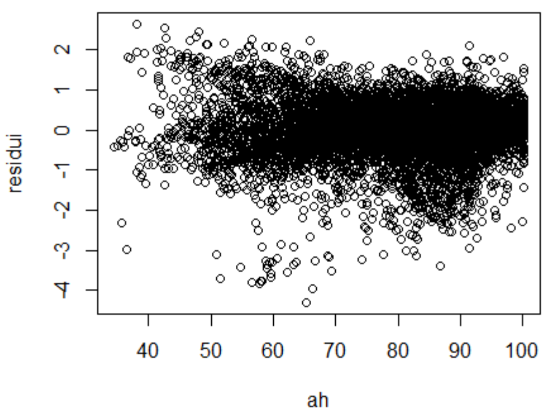
2. **Utilizzo dei test statistici di Breush-Pagan e White**: in questi test l'ipotesi nulla H_0 rappresenta l'ipotesi di omoschedasticità mentre l'ipotesi alternativa H_1 rappresenta l'allontanamento dall'omoschedasticità.

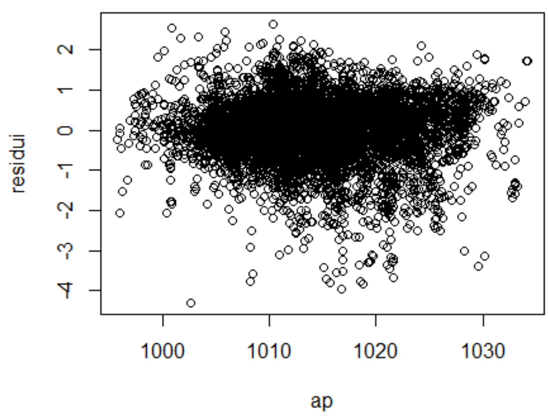
4.1 Analisi dei grafici di correlazione

I grafici da valutare rappresentano l'andamento dei residui stimati in funzione delle ordinate stimate e dei regressori:

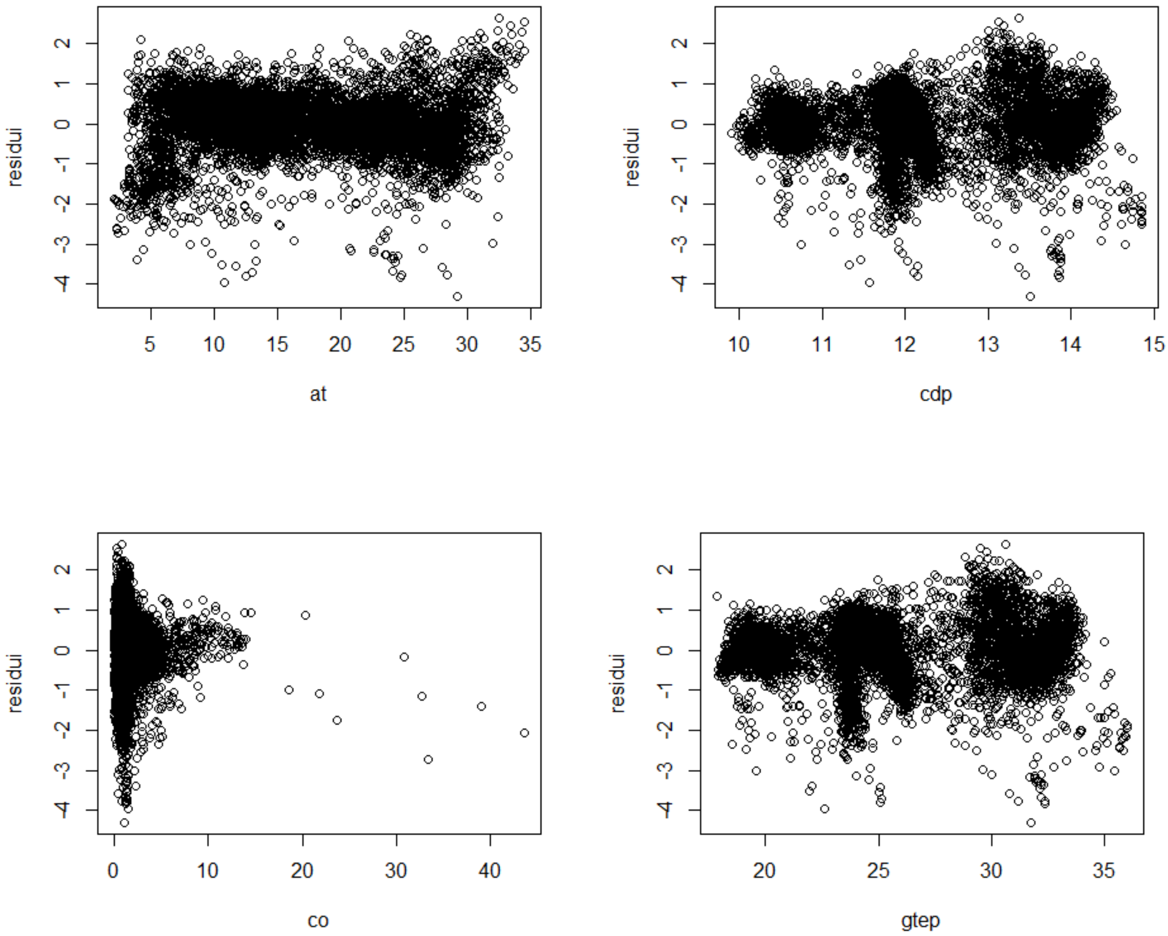








Si noti che:



- Il grafico (fit-residui) ha un andamento esplosivo;
- il grafico (afdp-residui) ha un andamento esplosivo-implosivo;
- il grafico (ah-residui) ha un andamento leggermente esplosivo;
- il grafico (ap-residui) non è informativo;
- il grafico (at-residui) non è informativo;
- il grafico (cdp-residui) non è informativo;
- il grafico (co-residui) è implosivo;
- il grafico (gtep-residui) non è informativo.

4.2 Test di Breush-Pagan

Il test di Breush Pagan consente di verificare se gli errori sono omoschedastici o meno. In termini formali abbiamo:

- H_0: omoschedasticità;
- H_1: ε^2 è in media funzione lineare dei regressori.

In questo primo test H_1 descrive l'ipotesi per la quale gli errori dipendano dai regressori, descritto formalmente tramite il seguente modello lineare ausiliario:

$$\hat{\varepsilon}^2{}_{\mathrm{i}} = \delta_0 + \delta_1 x_{1\mathrm{i}} + ... + \delta_{\mathrm{k}} x_{\mathrm{ki}} + u_{\mathrm{i}}$$

Questo implica che affinchè l'ipotesi nulla venga rifiutata deve esistere almeno un coefficiente del modello ausiliario diverso da zero. Il test BP è un test di Fisher sul modello lineare per gli errori appena descritto. Di conseguenza, se il $p - value < \alpha$ si rifiuta H_0, cioè si rifiuta che gli errori siano omoschedastici. L'implementazione è la seguente:

```
modelloStimato <- lm(TEY ~ AT+AP+AH+AFDP+GTEP+TIT+TAT+CDP+CO+NOX,dataset
   )
residui <- resid(modelloStimato)
residui_quadrato <- residui^2
modelloResiduiBP <- lm(residui_quadrato ~ at+ap+ah+afdp+gtep+tit+tat+cdp
   +co+nox)
summary(modelloResiduiBP)
```

```
Call:
lm(formula = residui_quadrato ~ at + ap + ah + afdp + gtep +
    tit + tat + cdp + co + nox)

Residuals:
    Min      1Q  Median      3Q     Max
-1.2718 -0.4881 -0.2259  0.1059 17.5666

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.115e+01  3.744e+00  -5.648 1.68e-08 ***
at          -7.218e-03  4.274e-03  -1.689 0.091252 .
ap           1.061e-02  3.050e-03   3.477 0.000509 ***
ah          -1.327e-02  1.368e-03  -9.702  < 2e-16 ***
afdp         4.788e-04  4.903e-02   0.010 0.992208
gtep         2.055e-01  7.024e-02   2.926 0.003448 **
tit          2.130e-02  1.492e-02   1.427 0.153558
tat         -1.080e-02  2.370e-02  -0.456 0.648702
cdp         -9.337e-01  1.983e-01  -4.710 2.53e-06 ***
co           2.533e-02  1.108e-02   2.286 0.022279 *
nox          1.411e-02  2.077e-03   6.794 1.18e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.103 on 7400 degrees of freedom
Multiple R-squared:  0.07215,	Adjusted R-squared:  0.0709
F-statistic: 57.55 on 10 and 7400 DF,  p-value: < 2.2e-16
```

E' possibile notare una relazione di minoranza tra il p-value e α della statistica F di Fisher, perciò si rifiuta l'ipotesi nulla.

4.3 Test di White

Analogamente al test BP, il test di White è un test di Fisher su un modello lineare per gli errori. Diversamente dal test Breush-Pagan, l'ipotesi alternativa di eteroschedasticità del Test di White riguarda la dipendenza dall'errore da una combinazione lineare delle ordinate stimate e i loro quadrati. Formalmente:

$$\hat{\varepsilon}^2{}_{\mathrm{i}} = \delta_0 + \delta_1 \hat{y}_{\mathrm{i}} + \delta_2 \hat{y}^2{}_{\mathrm{i}} + u_{\mathrm{i}}$$

Anche in questo caso l'ipotesi nulla è rifiutata se almeno uno dei coefficienti è diverso da zero. In R:

```
fit_quadrato <- fit^2
modelloResiduiW <-lm(residui_quadrato ~ fit+fit_quadrato)
summary(modelloResiduiW)
```

Anche in questo caso l'ipotesi nulla viene rifiutata.

```
Call:
lm(formula = residui_quadrato ~ fit + fit_quadrato)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.7332 -0.5244 -0.2412  0.0348 17.7486 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -6.348e+00  8.078e-01  -7.858 4.45e-15 ***
fit           9.031e-02  1.197e-02   7.546 5.04e-14 ***
fit_quadrato -2.879e-04  4.391e-05  -6.557 5.88e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.124 on 7408 degrees of freedom
Multiple R-squared:  0.03452,	Adjusted R-squared:  0.03426 
F-statistic: 132.4 on 2 and 7408 DF,  p-value: < 2.2e-16
```

4.4 Trasformazione del modello: divisione per le ordinate stimate

Proseguiamo con la trasformazione del modello lineare per ricondurlo ad un modello che rispetti l'ipotesi di omoschedasticità. La prima trasformazione consiste nel dividere il modello lineare per le ordinate stimate:

```
y<-tey/fit; x1 <- at/fit; x2<-ap/fit; x3<- ah/fit; x4<- afdp/fit; x5<-
    gtep/fit; x6 <- tit/fit; x7<- tat/fit;
x8 <- cdp/fit; x9 <- co/fit; x10 <- nox/fit;
reciproco_fit <- 1/fit

modelloTrovato <-lm(y~ reciproco_fit+x1+x2+x3+x4+x5+x6+x7+x8+x9+x10-1)
summary(modelloTrovato)
```

```
Call:
lm(formula = y ~ reciproco_fit + x1 + x2 + x3 + x4 + x5 + x6 + 
    x7 + x8 + x9 + x10 - 1)

Residuals:
       Min         1Q     Median         3Q        Max 
-0.0307828 -0.0027397  0.0004906  0.0032670  0.0181982 

Coefficients:
                Estimate Std. Error  t value Pr(>|t|)    
reciproco_fit -1.354e+02  2.300e+00  -58.895  < 2e-16 ***
x1            -3.527e-01  2.731e-03 -129.133  < 2e-16 ***
x2            -6.475e-02  1.954e-03  -33.140  < 2e-16 ***
x3            -6.813e-03  8.546e-04   -7.972 1.79e-15 ***
x4            -4.257e-01  3.212e-02  -13.252  < 2e-16 ***
x5             3.133e-01  4.792e-02    6.538 6.65e-11 ***
x6             6.284e-01  9.803e-03   64.105  < 2e-16 ***
x7            -6.562e-01  1.561e-02  -42.025  < 2e-16 ***
x8             1.131e+00  1.226e-01    9.226  < 2e-16 ***
x9             1.278e-02  6.338e-03    2.017   0.0438 *  
x10           -1.564e-02  1.204e-03  -12.991  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.005211 on 7400 degrees of freedom
Multiple R-squared:      1,	Adjusted R-squared:      1 
F-statistic: 2.481e+07 on 11 and 7400 DF,  p-value: < 2.2e-16
```

Applichiamo il Test di Breush Pagan al nuovo modello per vedere se è stata rimossa l'eteroschedasticità:

```
residuiTrovato <- resid(modelloTrovato)
residuiTrovato_quadrato <- residuiTrovato^2
newModelBP <- lm(residuiTrovato_quadrato ~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10
    -1)
summary(newModelBP)
```

```
Call:
lm(formula = residuiTrovato_quadrato ~ x1 + x2 + x3 + x4 + x5 +
    x6 + x7 + x8 + x9 + x10 - 1)

Residuals:
       Min         1Q     Median         3Q        Max
-7.388e-05 -2.469e-05 -1.236e-05  5.020e-06  9.026e-04

Coefficients:
      Estimate Std. Error t value Pr(>|t|)
x1  -1.133e-04  2.896e-05  -3.913 9.20e-05 ***
x2  -3.605e-05  1.210e-05  -2.980  0.00289 **
x3  -1.062e-04  8.908e-06 -11.927  < 2e-16 ***
x4   5.858e-06  3.400e-04   0.017  0.98625
x5   5.275e-04  4.730e-04   1.115  0.26473
x6   1.536e-04  1.016e-04   1.512  0.13066
x7  -1.620e-04  1.575e-04  -1.028  0.30380
x8  -3.897e-03  1.295e-03  -3.010  0.00262 **
x9   6.056e-05  6.369e-05   0.951  0.34173
x10  9.180e-05  1.276e-05   7.197 6.77e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.526e-05 on 7401 degrees of freedom
Multiple R-squared:  0.2322,    Adjusted R-squared:  0.2311
F-statistic: 223.8 on 10 and 7401 DF,  p-value: < 2.2e-16
```

Anche in questo caso si rifiuta l'ipotesi nulla, in quanto il p-value è minore di 0.05.

4.5 Trasformazione logaritmica della variabile dipendente

Applichiamo il logaritmo alla variabile dipendente:

```
logTey <-log(tey)
modelloLogaritmico <- lm(logTey~at+ap+ah+afdp+gtep+tit+tat+cdp+co+nox)
summary(modelloLogaritmico)
```

```
Call:
lm(formula = logTey ~ at + ap + ah + afdp + gtep + tit + tat +
    cdp + co + nox)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0305281 -0.0032744  0.0003953  0.0036942  0.0252713

Coefficients:
              Estimate Std. Error  t value Pr(>|t|)
(Intercept)  1.852e+00  2.046e-02   90.507  < 2e-16 ***
at          -2.472e-03  2.335e-05 -105.838  < 2e-16 ***
ap          -5.007e-04  1.667e-05  -30.044  < 2e-16 ***
ah          -1.072e-05  7.474e-06   -1.434   0.1516
afdp        -3.559e-03  2.680e-04  -13.281  < 2e-16 ***
gtep         9.790e-04  3.839e-04    2.550   0.0108 *
tit          5.354e-03  8.155e-05   65.654  < 2e-16 ***
tat         -4.284e-03  1.295e-04  -33.085  < 2e-16 ***
cdp          9.076e-03  1.083e-03    8.377  < 2e-16 ***
co           5.421e-04  6.056e-05    8.953  < 2e-16 ***
nox          8.630e-05  1.135e-05    7.603 3.26e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.006026 on 7400 degrees of freedom
Multiple R-squared:  0.9975,    Adjusted R-squared:  0.9975
F-statistic: 2.971e+05 on 10 and 7400 DF,  p-value: < 2.2e-16
```

Applichiamo nuovamente il Test di Breush Pagan:

```
residui_logaritmo <- resid(modelloLogaritmico)
residui_logaritmo_quadrato <- residui_logaritmo^2
modLogBP <- lm(residui_logaritmo_quadrato ~ at+ap+ah+afdp+gtep+tit+tat+
    cdp+co+nox)
summary(modLogBP)
```

```
Call:
lm(formula = residui_logaritmo_quadrato ~ at + ap + ah + afdp +
    gtep + tit + tat + cdp + co + nox)

Residuals:
       Min         1Q     Median         3Q        Max
-1.848e-04 -3.022e-05 -1.326e-05  6.550e-06  8.610e-04

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.803e-04  2.199e-04  -1.275  0.20242
at           1.227e-06  2.510e-07   4.890 1.03e-06 ***
ap           1.027e-06  1.791e-07   5.734 1.02e-08 ***
ah          -6.881e-07  8.032e-08  -8.566  < 2e-16 ***
afdp        -8.226e-06  2.880e-06  -2.856  0.00430 **
gtep         2.263e-05  4.126e-06   5.484 4.29e-08 ***
tit         -2.159e-06  8.764e-07  -2.464  0.01377 *
tat          2.521e-06  1.392e-06   1.812  0.07008 .
cdp         -3.608e-05  1.164e-05  -3.099  0.00195 **
co           2.059e-07  6.508e-07   0.316  0.75176
nox          2.528e-06  1.220e-07  20.722  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.477e-05 on 7400 degrees of freedom
Multiple R-squared:  0.1216,    Adjusted R-squared:  0.1204
F-statistic: 102.4 on 10 and 7400 DF,  p-value: < 2.2e-16
```

Il p-value risulta essere minore di 0.05, perciò anche con questa trasformazione non è stata rimossa l'eteroschedasticità.

4.6 Trasformazione LOG-LOG

Un'altra trasformazione tentata riguarda quella LOG-LOG:

```
log_tey <- log(tey)
log_at <- log(at)
log_ap <- log(ap)
log_ah <- log(ah)
log_afdp<- log(afdp)
log_gtep <- log(gtep)
log_tit <- log(tit)
log_tat <- log(tat)
log_cdp <- log(cdp)
log_co <- log(co)
log_nox<- log(nox)

modello_log_log <- lm(log_tey~log_at+log_ap+log_ah+log_afdp+log_gtep+log
    _tit+log_tat+log_cdp+log_co+log_nox)
summary(modello_log_log)
```

Si applica nuovamente il Test BP:

```
residui_log_log<- resid(modello_log_log)
residui_log_log_quadrato <- residui_log_log^2
mod_log_log_BP <- lm(residui_log_log_quadrato ~ log_at+log_ap+log_ah+log
    _afdp+log_gtep+log_tit+log_tat+log_cdp+log_co+log_nox)
summary(mod_log_log_BP)
```

Anche in quest'ultimo caso non è riuscita la rimozione dell'eteroschedasticità.

4.7 Divisione per i regressori

Effettuiamo infine la tarsformazione del modello dividendolo per i regressori. Anche in questo caso non si è riusciti a rimuovere l'eteroschedasticità. Di seguito è riportata l'applicazione del

```
Call:
lm(formula = log_tey ~ log_at + log_ap + log_ah + log_afdp +
    log_gtep + log_tit + log_tat + log_cdp + log_co + log_nox)

Residuals:
      Min        1Q    Median        3Q       Max
-0.037454 -0.003920  0.000755  0.004915  0.025756

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.525e+01  2.350e-01 -64.890  < 2e-16 ***
log_at      -2.526e-02  3.415e-04 -73.975  < 2e-16 ***
log_ap      -3.488e-01  1.973e-02 -17.681  < 2e-16 ***
log_ah       1.757e-02  5.621e-04  31.261  < 2e-16 ***
log_afdp    -5.509e-03  1.380e-03  -3.991 6.65e-05 ***
log_gtep     4.190e-02  1.058e-02   3.961 7.54e-05 ***
log_tit      4.999e+00  9.429e-02  53.013  < 2e-16 ***
log_tat     -2.088e+00  6.134e-02 -34.045  < 2e-16 ***
log_cdp      2.233e-01  1.503e-02  14.857  < 2e-16 ***
log_co       1.406e-03  1.464e-04   9.606  < 2e-16 ***
log_nox      2.225e-02  8.767e-04  25.375  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.007288 on 7400 degrees of freedom
Multiple R-squared:  0.9964,	Adjusted R-squared:  0.9964
F-statistic: 2.029e+05 on 10 and 7400 DF,  p-value: < 2.2e-16
```

```
Call:
lm(formula = residui_log_log_quadrato ~ log_at + log_ap + log_ah +
    log_afdp + log_gtep + log_tit + log_tat + log_cdp + log_co +
    log_nox)

Residuals:
       Min         1Q     Median         3Q        Max
-1.494e-04 -4.124e-05 -1.847e-05  1.143e-05  1.279e-03

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.506e-02  2.919e-03  -5.161 2.52e-07 ***
log_at      -6.393e-05  4.241e-06 -15.075  < 2e-16 ***
log_ap      -3.881e-04  2.450e-04  -1.584   0.1133
log_ah      -1.397e-04  6.982e-06 -20.003  < 2e-16 ***
log_afdp    -8.540e-05  1.715e-05  -4.981 6.47e-07 ***
log_gtep     2.171e-04  1.314e-04   1.652   0.0985 .
log_tit      4.655e-03  1.171e-03   3.974 7.12e-05 ***
log_tat     -1.871e-03  7.618e-04  -2.456   0.0141 *
log_cdp     -1.206e-03  1.867e-04  -6.458 1.13e-10 ***
log_co       3.537e-06  1.818e-06   1.945   0.0518 .
log_nox      6.465e-05  1.089e-05   5.937 3.04e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.052e-05 on 7400 degrees of freedom
Multiple R-squared:  0.1193,	Adjusted R-squared:  0.1181
F-statistic: 100.2 on 10 and 7400 DF,  p-value: < 2.2e-16
```

Test di Breush Pagan per il modello trasformato dividendo il modello originale per il regressore CDP. I risultati ottenuti con gli altri modelli sono analoghi:

```r
y_cdp<-tey/cdp; x1_cdp <- at/cdp; x2_cdp<-ap/cdp; x3_cdp<- ah/cdp; x4_cdp<- afdp/cdp; x5_cdp<- gtep/cdp; x6_cdp <- tit/cdp; x7_cdp<- tat/cdp;
x8_cdp <- cdp/cdp; x9_cdp <- co/cdp; x10_cdp <- nox/cdp;
reciproco_cdp <- 1/cdp

modelloTrovato_cdp <-lm(y~ reciproco_cdp+x1_cdp+x2_cdp+x3_cdp+x4_cdp+x5_cdp+x6_cdp+x7_cdp+x8_cdp+x9_cdp+x10_cdp-1)
summary(modelloTrovato_cdp)

#Ripetizione Test Breush Pagan

residuiTrovato_cdp <- resid(modelloTrovato_cdp)
residuiTrovato_quadrato_cdp <- residuiTrovato_cdp^2
```

```
newModelBP_cdp <- lm(residuiTrovato_quadrato_cdp ~x1_cdp+x2_cdp+x3_cdp+
   x4_cdp+x5_cdp+x6_cdp+x7_cdp+x8_cdp+x9_cdp+x10_cdp-1)
summary(newModelBP_cdp)
```

```
Call:
lm(formula = residuiTrovato_quadrato_cdp ~ x1_cdp + x2_cdp +
    x3_cdp + x4_cdp + x5_cdp + x6_cdp + x7_cdp + x8_cdp + x9_cdp +
    x10_cdp - 1)

Residuals:
       Min         1Q     Median         3Q        Max
-7.541e-05 -2.458e-05 -1.217e-05  5.020e-06  9.011e-04

Coefficients:
          Estimate Std. Error t value Pr(>|t|)
x1_cdp  -8.667e-06  2.618e-06  -3.310 0.000936 ***
x2_cdp  -3.026e-06  1.101e-06  -2.749 0.006001 **
x3_cdp  -9.613e-06  8.100e-07 -11.869  < 2e-16 ***
x4_cdp   2.433e-07  3.034e-05   0.008 0.993603
x5_cdp   4.550e-05  4.214e-05   1.080 0.280255
x6_cdp   1.372e-05  9.122e-06   1.504 0.132753
x7_cdp  -1.487e-05  1.414e-05  -1.052 0.292928
x8_cdp  -3.487e-04  1.175e-04  -2.968 0.003011 **
x9_cdp   5.897e-06  5.949e-06   0.991 0.321554
x10_cdp  9.075e-06  1.184e-06   7.662 2.06e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.514e-05 on 7401 degrees of freedom
Multiple R-squared:  0.2338,    Adjusted R-squared:  0.2327
F-statistic: 225.8 on 10 and 7401 DF,  p-value: < 2.2e-16
```

Anche in quest'ultimo caso non si è ottenuto un p-value minore di 0.05.

5 *Regolarizzazione e modelli di apprendimento*

In quest'ultima sezione della relazione è stata tralasciata la trattazione del modello di regressione da una prospettiva inferenziale per concentrarsi maggiormente su un suo utilizzo di tipo previsivo: tale approccio implica un contesto per il quale è si possibile osservare il vettore di regressori X ma la corrispondente osservazione sulla variabile dipendete Y risulta impossibile e/o sconsigliata.

Considerata l'impossibilità di determinare in maniera assoluta quale sia il metodo previsivo migliore ci si è concentrati sul confronto tra diverse tecniche al termine del quale si prenderà in considerazione la strategia che ha restituito il più piccolo Errore Quadratico Medio, denotazione della sua capacità di fornire le previsioni più accurate possibili.
In particolare per valutare l'accuratezza delle previsioni si suddivide l'insieme delle n osservazioni in due set di dati: 1) training set (con selezione casuale delle unità) su cui si costruisce il modello di apprendimento; 2) test set (composto dalle restanti unità), utile per la misurazione delle performance in termini predittivi su un data set mai utilizzato. L'idea è quella di giudicare l'errore quadratico medio associato al test set: se piccolo si concluderà che il modello di apprendimento fornisce previsioni accurate.

Solitamente la suddivisione del data set viene effettuata allocando il 70% delle osservazioni nel training set e il 30% nel test set. Considerato che l'efficacia del metodo di apprendimento può dipendere dalla specifica suddivisione del dataset originario si utilizzano in generale due tecniche di ricampionamento:

1. **K-fold Cross Validation**
    - Il dataset viene casualmente suddiviso in k gruppi di dimensioni approssimativamente equivalenti.
    - Il primo gruppo è utilizzato come test set, mentre il modello di apprendimento viene addestrato sui restanti k-1 gruppi.
    - Questa procedura viene ripetuta k volte, con ciascun gruppo utilizzato almeno una volta come test set.
    - Alla fine, si ottiene una stima più robusta delle performance del modello, dato che tutte le osservazioni sono state utilizzate sia per il training che per il testing.
2. **Leave-One-Out Cross Validation**
    - In questa tecnica il test set è composto da una singola osservazione mentre le rimanenti osservazioni costituiscono il training set.
    - Questa procedura viene ripetuta tante volte quanti sono gli elementi nel dataset, con ogni osservazione che agisce una volta come parte del test set.
    - Questo approccio offre una validazione incrociata molto dettagliata, ma può essere computazionalmente costoso.

Prima di passare alle diverse tecniche di regolarizzazione è necessaria un'operazione di preprocessing sui dati, la standardizzazione: questa si rende utile nei casi in cui i regressori sono espressi in unità e ordini di grandezza differenti.
Questo tipo di trasformazione ci consentirà di esprimere tutte le osservazioni come numeri adimensionali.

Nel dataset preso in considerazione l'operazione di standardizzazione è necessaria su quasi tutti i regressori presenti tranne AH (umidità ambientale) già di per se adimensionale:

1. AT: temperatura ambientale (°C);
2. AP: pressione ambientale (mbar);

3. AFDP: differenza di pressione del filtro dell'aria (mbar);
4. GTEP: pressione di scarico della turbina a gas (mbar);
5. TIT: temperatura in ingresso della turbina (°C);
6. TAT: temperatura in uscita della turbina (°C);
7. CDP: pressione di scarico del compressore (mbar);
8. CO: monossido di carbonio (mg/m3);
9. NOX: ossido d'azoto (mg/m3);

```
dati <- read.csv(file_path, header = TRUE, dec = ".", sep = ",")
set.seed(100)

head(dati)

summary(dati)

AT_STD <- scale(dati$AT)
AP_STD <- scale(dati$AP)
AH <- dati$AH #Non c'  bisogno di standardizzare poich  un valore percentuale
AFDP_STD <- scale(dati$AFDP)
GTEP_STD <- scale(dati$GTEP)
TIT_STD <- scale(dati$TIT)
TAT_STD <- scale(dati$TAT)
TEY<- dati$TEY
CDP_STD <- scale(dati$CDP)
CO_STD <- scale(dati$CO)
NOX_STD <- scale(dati$NOX)

matriceDati_STD <- cbind(AT_STD,AP_STD,AH,AFDP_STD,GTEP_STD,TIT_STD,TAT_STD,TEY,CDP_STD,CO_STD,NOX_STD)
colnames(matriceDati_STD) <- c("AT_STD","AP_STD","AH","AFDP_STD","GTEP_STD","TIT_STD","TAT_STD","TEY","CDP_STD","CO_STD","NOX_STD")

dati_STD <- as.data.frame(matriceDati_STD)

head(dati_STD)

#Stima del primo modello di regressione lineare
modello_lineare <- lm(TEY~AT_STD+AP_STD+AH+AFDP_STD+GTEP_STD+TIT_STD+TAT_STD+CDP_STD+CO_STD+NOX_STD, data = dati_STD)
summary(modello_lineare) #N.B:  stato ottenuto lo stesso modello di stima nella fase di inferenza
```

5.1 Ridge Regression

Tale strategia di regolarizzazione si concretizza tramite la minimizzazione della seguente funzione di perdita:

$$L_{\text{ridge}}(\beta;\lambda) = (Y - X\beta)^{\text{t}}(Y - X\beta) + \lambda * \sum_{j=1}^{k} {\beta_{\text{j}}}^2$$

dove $(Y - X\beta)^{\text{t}}(Y - X\beta)$ è la somma dei quadrati degli errori mentre $\lambda * \sum_{j=1}^{k} \beta^2{}_{\text{j}}$ è il termine di penalizzazione, misura della complessità del vettore dei coefficenti di regressione. L'obiettivo sarà quello di individuare il *ridge regression estimator* ossia lo stimatore di tali coefficenti che minimizza la funzione di perdita; per questa operazione sarà necessario considerare un insieme di valori lambda (parametro di penalità) da testare. In questo contesto la particolarità della Ridge Regression sta proprio nell'idea che all'aumentare del valore lambda i coefficenti di regressione tendono a 0 senza mai raggiungere l'uguaglianza.

```
1  X_Regressori <- as.matrix(dati_STD[,-8])
2  Y_Tey <- dati_STD[,8]
3  nRighe <- nrow(dati_STD)
4
5  valoriLambda <- 10^seq(8,-4,length = 100)
6
7  #----RIDGE REGRESSION----
8
9  all_modelli_ridge <- glmnet(X_Regressori,Y_Tey,lambda = valoriLambda,
      alpha = 0, standardize = FALSE)
10 coef(all_modelli_ridge)
11
12 plot(all_modelli_ridge,xvar = "lambda",label = TRUE)
13 title(main = "Ridge Regression. No STD", line = 2.5)
```

Tramite l'utilizzo dei comandi a riga 9 e 10 è possibile riportare rispettivamente i modelli di previsione ottenuti tramite ridge regression e le stime dei coefficenti acquisite tramite i diversi ridge regression estimator (uno per ogni lambda presente nell'insieme scelto).

Di seguito il grafico che mostra l'andamento dei coefficenti all'aumentare del parametro lambda:

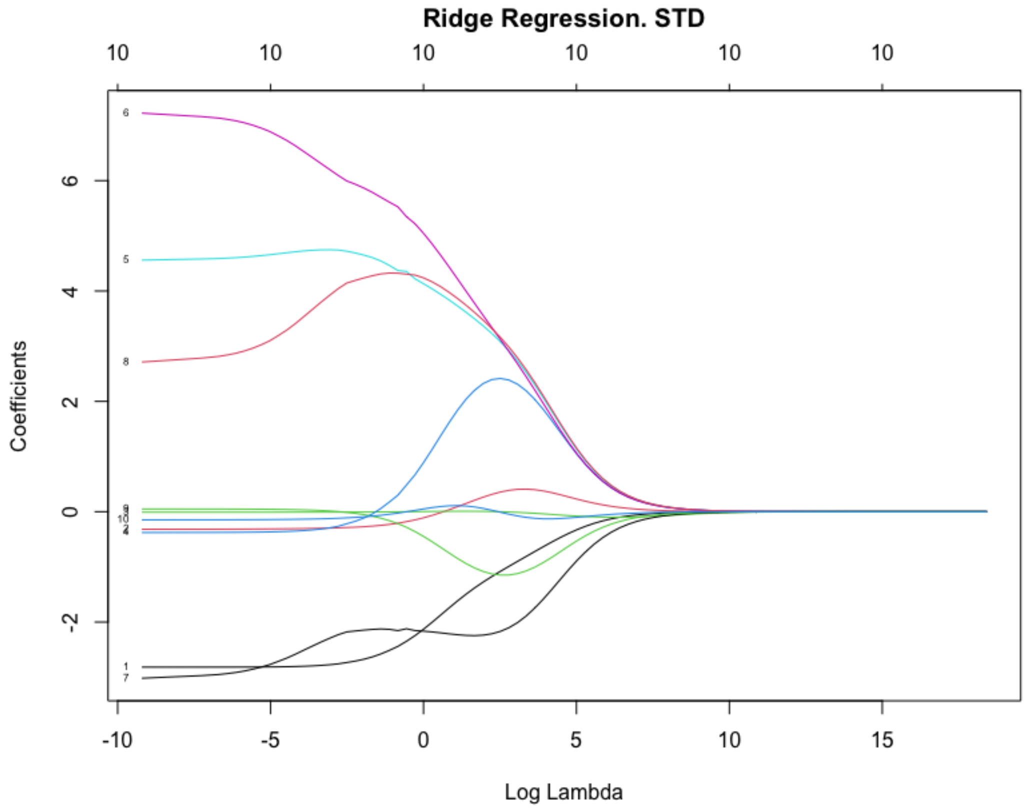


Figura 3: *È possibile notare come all'aumentare del parametro di penalità i coefficenti di regressioni vengano man mano azzerati*

5.1.1 K-Fold Cross Validation: K=10

Eseguiamo a questo punto la prima tecnica di ricampionamento del dataset:

```
1 all_modelli_ridge_CVK10 <- cv.glmnet(X_Regressori,Y_Tey,lambda = valoriLambda, alpha = 0) # K=10
2 plot(all_modelli_ridge_CVK10)
3 title(main = "Ridge Regression: K-Fold (K=10)", line = 2.5)
4 
5 #Lambda min
6 lmin_K10 <- all_modelli_ridge_CVK10$lambda.min
7 lmin_K10
8 
9 #Costruisco il modello di previsione in funzione del lambda min
10 modello_ridge_lminK10 <- glmnet(X_Regressori,Y_Tey,lambda = lmin_K10, alpha = 0,standardize = FALSE)
11 
12 #Parametri/Coefficenti di regressione del modello
13 coef_modello_ridge_lminK10 <- coef(modello_ridge_lminK10)[,1]
14 coef_modello_ridge_lminK10
15 
16 #MSE_min del modello
17 mseMin_lminK10 <- all_modelli_ridge_CVK10$cvm[all_modelli_ridge_CVK10$lambda == all_modelli_ridge_CVK10$lambda.min]
18 mseMin_lminK10
```

È importante appuntare alcune cose riguardo questa sezione di codice.

Riga 1: Il comando *cv.glmnet* ha reso possibile la stima di un gruppo di modelli previsivi in funzione dell'insieme di valori lambda scelto in precedenza (*valoriLambda*): su ognuno di questi modelli è stata successivamente applicata la medesima tecnica di ricampionamento K-Fold introdotta all'inizio del paragrafo [1].

Riga 6-10: Il comando glmnet restituisce in definitiva il modello di previsione migliore in grado di minimizzare l'MSE sfruttando il più piccolo valore lambda estratto.

Riga 13-14: Stampa dei coefficenti di regressioni associati al migliore modello previsivo ottenuto;

Riga 17: Restituzione dell'MSE del modello previsivo ottenuto.

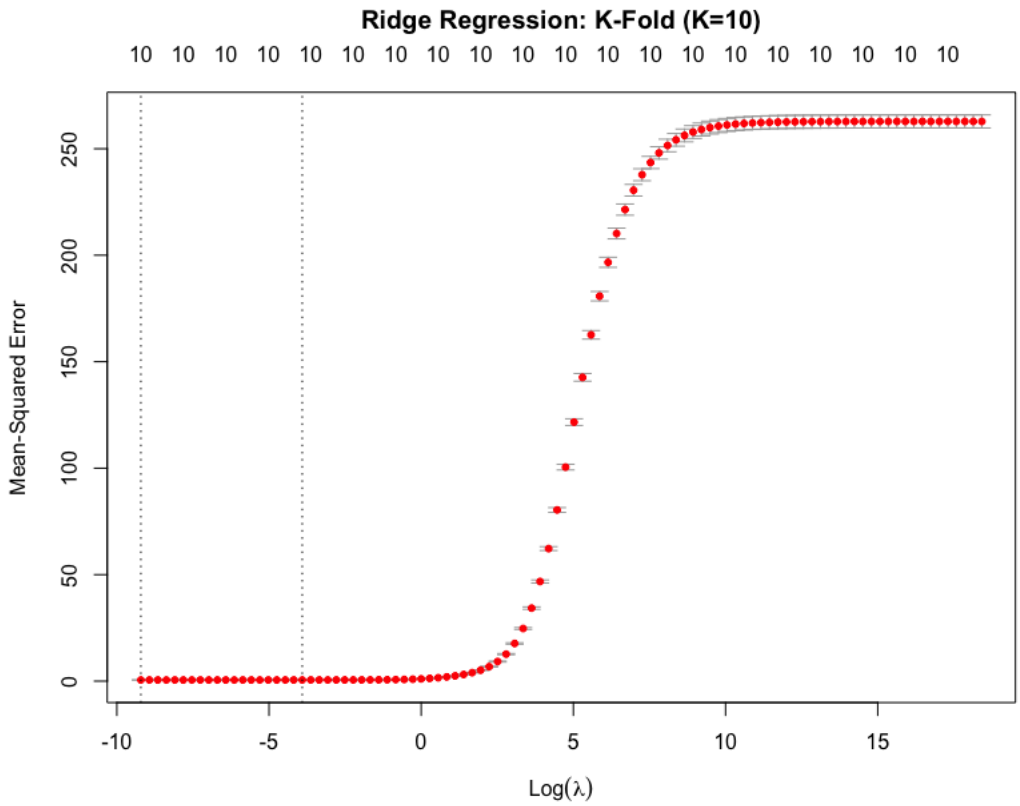
I principali risultati ottenuti in questa sezione sono quindi:

1. $\lambda_{\min} = 1e - 04$
2. $\hat{\beta}(\lambda_{\min}) =$

```
  (Intercept)        AT_STD        AP_STD            AH      AFDP_STD      GTEP_STD       TIT_STD
136.456149993  -2.763482517  -0.361394119  -0.008973473  -0.369852180   3.353575022   9.149176247
      TAT_STD       CDP_STD        CO_STD       NOX_STD
 -4.535801698   0.978463215   0.050115533  -0.164192935
```

3. $\text{MSE}_{\min} = \mathbf{0.555384}$.

Nel grafico possiamo notare come varia l'MSE al variare del parametro di lambda; la peculiarità del piano in questione è l'invisibilità ad occhio nudo del punto di flesso in cui si incrociano l'MSE e il valore di lambda che lo minimizza.



5.1.2 Funzione "calcola-CV-MSE-Modulare"

Prima di andare avanti nella relazione dell'elaborato si è ritenuta necessaria la definizione di una funzione di utilità che permettesse un processo di stima nonché una valutazione più veloce dei modelli di previsione ottenuti tramite le diverse tecniche di regolarizzazione e cross-validation.

```
calcola_CV_MSE_Modulare <- function(X,Y,k,a){

  lambdaValues <- 10^seq(8,-4,length = 100)

  modelliRidge_CV <- cv.glmnet(X,Y,lambda = lambdaValues,alpha = a, nfolds = k)

  if (a == 0){
    plot(modelliRidge_CV)
    titolo <- sprintf("Ridge Regression: K-Fold (K=%d). STD", k)
    title(main = titolo, line = 2.5)
  }
  if (a == 1){
    plot(modelliRidge_CV)
    titolo <- sprintf("Lasso Regression: K-Fold (K=%d). STD", k)
    title(main = titolo, line = 2.5)
  }
  if (a > 0 && a < 1){
    plot(modelliRidge_CV)
    titolo <- sprintf("Elastic Net (alpha = %g): K-Fold (K=%g). STD",a,k)
```

```
    title(main = titolo, line = 2.5)
  }


  lmin <- modelliRidge_CV$lambda.min

  modelloRidge_CV_LMIN <- glmnet(X,Y,lambda = lmin, alpha = a, standardize = FALSE)

  coef_modelloRidge_CV_LMIN <- coef(modelloRidge_CV_LMIN)[,1]

  MSE_min <- modelliRidge_CV$cvm[modelliRidge_CV$lambda == modelliRidge_CV$lambda.min]

  risultato <- list(modello = modelloRidge_CV_LMIN, mse = MSE_min, lam = lmin, coefficenti = coef_modelloRidge_CV_LMIN)

  return(risultato)

}
```

La funzione è caratterizzata da 4 parametri di input:

1. **X**: matrice dei regressori;
2. **Y**: vettore delle variabili dipendenti;
3. **k**: parametro di cross-validation;
4. **a**: parametro di regolarizzazione;

In base alle diverse combinazioni di k e a, la funzione calcolerà e restituirà un vettore contenente:

1. **Modello**: migliore modello previsivo ottenuto combinando la tecnica di regolarizzazione con coefficente a e ricampionamento k;
2. **MSE**: errore quadratico medio associato al modello previsivo;
3. **Lmin**: valore di lambda che minimizza l'errore quadratico medio del modello;
4. **Coefficenti**: coefficenti β del modello.

Il suo utilizzo ci permetterà di evitare la ripetizione delle stesse righe di codice in casi differenti all'interno del progetto migliorandone la leggibilità e la comprensione.

5.1.3 K-Fold Cross Validation: K=5

```
listaRes_RIDGE_K5 <- calcola_CV_MSE_Modulare(X_Regressori,Y_Tey,5,0)
#1 lmin_RIDGE_K5 <- listaRes_RIDGE_K5$lam
#2 coefs_RIDGE_K5 <- listaRes_RIDGE_K5$coefficenti
#3 mse_RIDGE_K5 <- listaRes_RIDGE_K5$mse
```

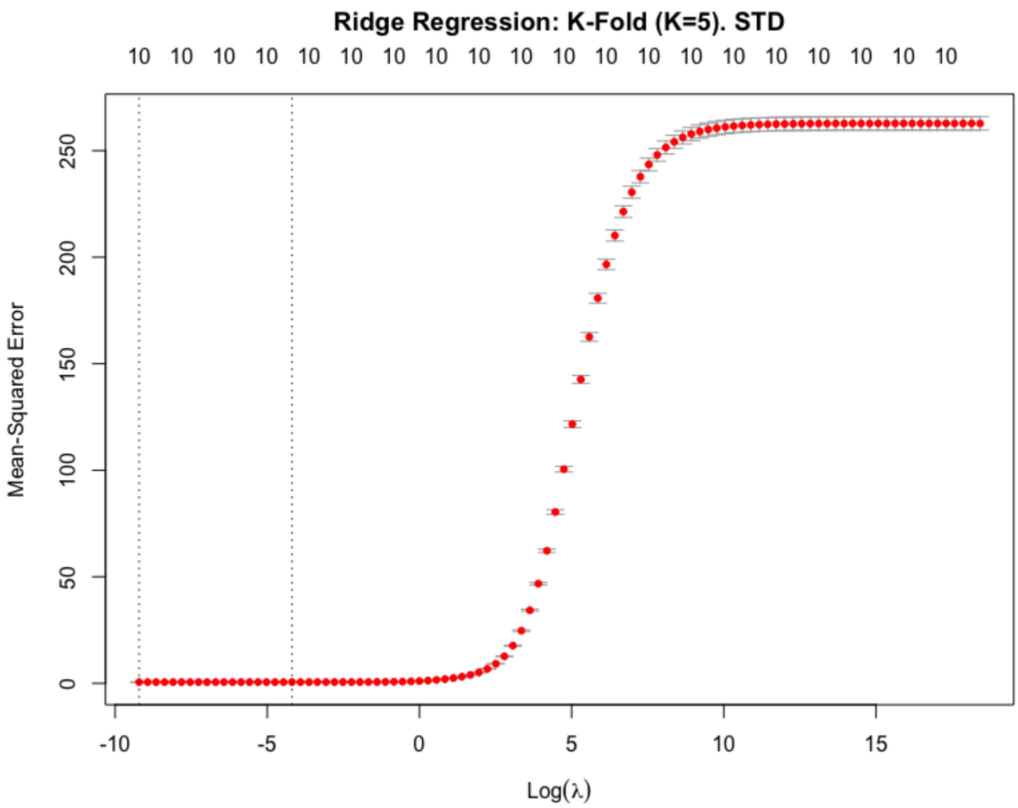
I principali risultati ottenuti sono:

1. $\lambda_{\min} = 1e - 04$
2. $\hat{\beta}(\lambda_{\min}) =$

```
  (Intercept)       AT_STD       AP_STD           AH     AFDP_STD     GTEP_STD      TIT_STD
136.456149993 -2.763482517 -0.361394119 -0.008973473 -0.369852180  3.353575022  9.149176247
      TAT_STD      CDP_STD       CO_STD      NOX_STD
 -4.535801698  0.978463215  0.050115533 -0.164192935
```

3. $\text{MSE}_{\min}$ = **0.554795**.

Il grafico sarà il seguente:



Da questo primo confronto è possibile notare come a parità di tecnica di regolarizzazione la scelta di una cross-validation di tipo K-Fold con k=5 piuttosto che con K=10 ci permette di ottenere un modello previsivo con una bontà di previsione leggermente migliore: **0.554795** del modello 5-Fold contro **0.555384** del modello 10-Fold.

5.2 Lasso

Concretizzata tramite la minimizzazione della seguente funzione di perdita:

$$L_{\text{lasso}}(\beta; \lambda) = (Y - X\beta)^{\text{t}}(Y - X\beta) + \lambda * \sum_{j=1}^{k} |\beta_{\text{j}}|$$

La caratteristica principale della tecnica Lasso nonché ciò che la differenzia dalla Ridge Regression è il completo annullamento, piuttosto che tendenza a 0, di alcuni coefficenti di regressione in corrispondenza di determinati valori del parametro di penalità λ.

Inoltre, per valori prossimi allo zero del parametro λ lo stimatore Lasso è prossimo allo stimatore ai minimi quadrati mentre per valori molto elevati di λ il termine di penalità tende a mascherare la somma dei quadrati. Il seguente grafico mostra l'andamento dei coefficenti all'aumentare del parametro lambda nel nostro contesto di applicazione:

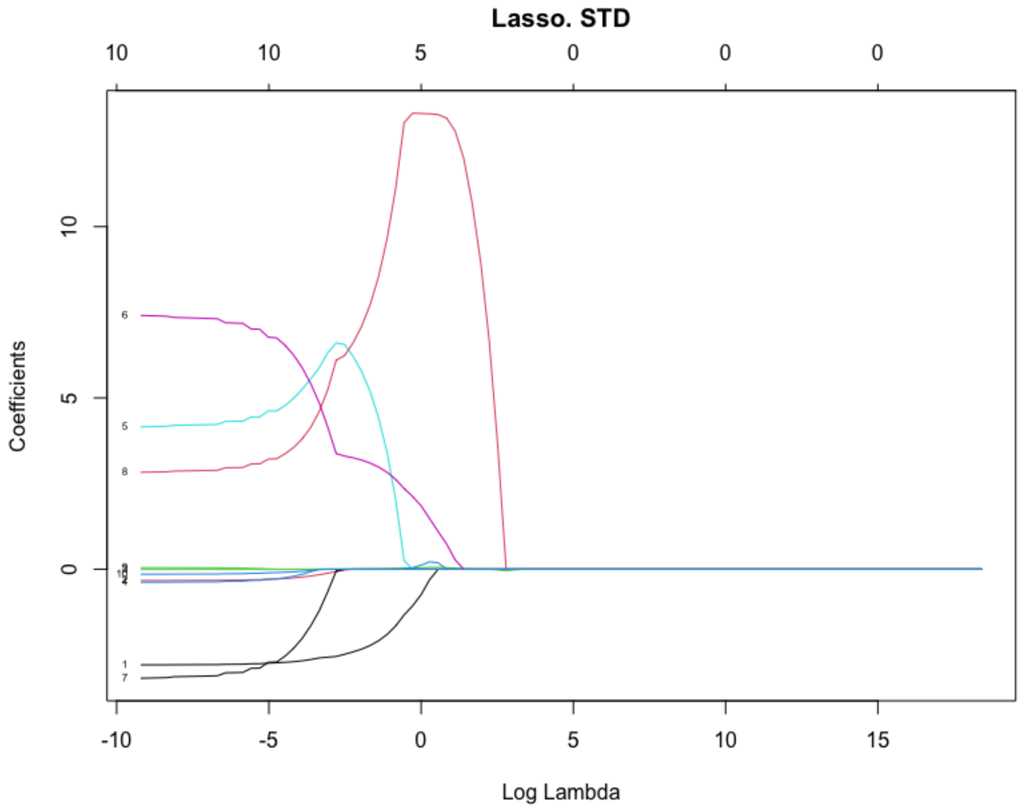


Figura 4: In questo caso i β_{j} che vengono azzerati sono quelli tali per cui $|\beta_{\text{j}}| \leq \frac{1}{2} * \lambda$ ovvero quei coefficenti il cui valore massimo non ha superato il **valore di soglia** $\frac{1}{2} * \lambda$.

5.2.1 K-Fold Cross Validation: K = 10

Come nel caso della tecnica di regolarizzazione precedente si procede per convenzione con l'applicazione del processo di cross-validation di tipo 10-Fold:

```
listaRes_LASSO_K10 <- calcola_CV_MSE_Modulare(X_Regressori,Y_Tey,10,1)
mse_LASSO_K10 <- listaRes_LASSO_K10$mse
lmin_LASSO_K10 <- listaRes_LASSO_K10$lam
coefs_LASSO_K10 <- listaRes_LASSO_K10$coefficenti
```

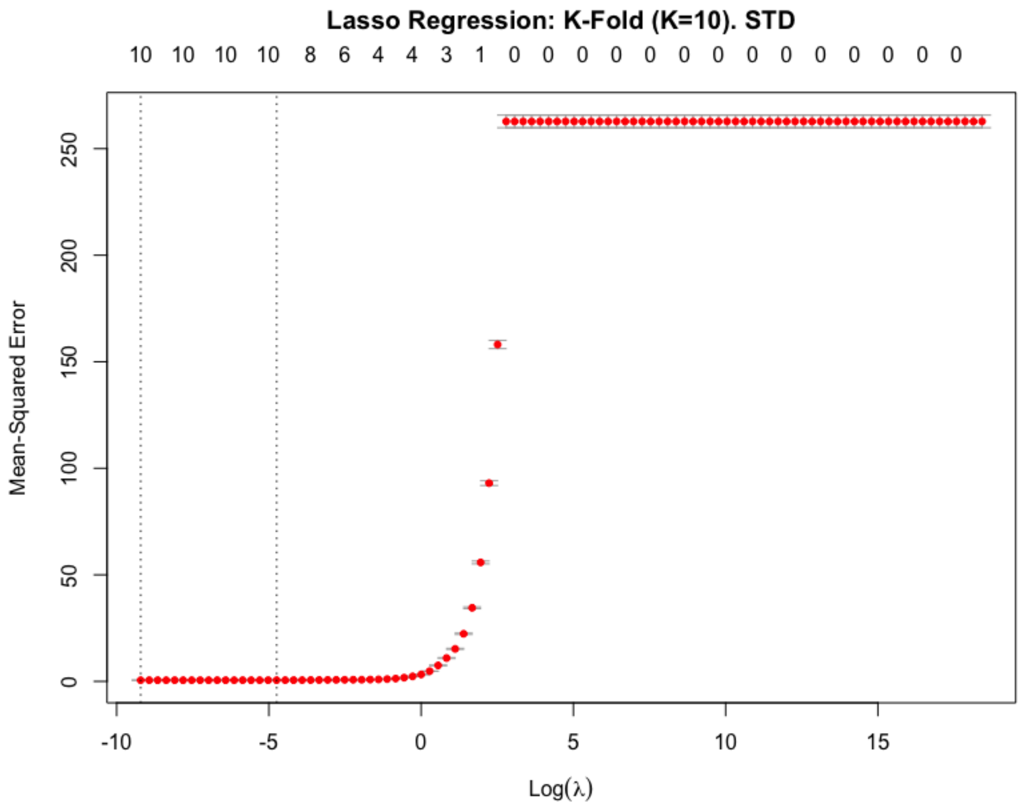
In questo caso i principali risultati ottenuti sono:

1. $\lambda_{\text{min}} = 1e - 04$
2. $\hat{\beta}(\lambda_{\text{min}}) =$

```
  (Intercept)        AT_STD        AP_STD            AH      AFDP_STD      GTEP_STD       TIT_STD
136.454316660  -2.762367391  -0.361258171  -0.008950318  -0.368627944   3.348456485   9.155765880
      TAT_STD       CDP_STD        CO_STD       NOX_STD
 -4.541496738   0.971898019   0.049602680  -0.163793650
```

3. $\text{MSE}_{\text{min}} =$ **0.551349**.

Con il grafico:



5.2.2 K-Fold Cross Validation: K=5

```
listaRes_LASSO_K5 <- calcola_CV_MSE_Modulare(X_Regressori,Y_Tey,5,1)
mse_LASSO_K5 <- listaRes_LASSO_K5$mse
lmin_LASSO_K5 <- listaRes_LASSO_K5$lam
coefs_LASSO_K5 <- listaRes_LASSO_K5$coefficenti
```

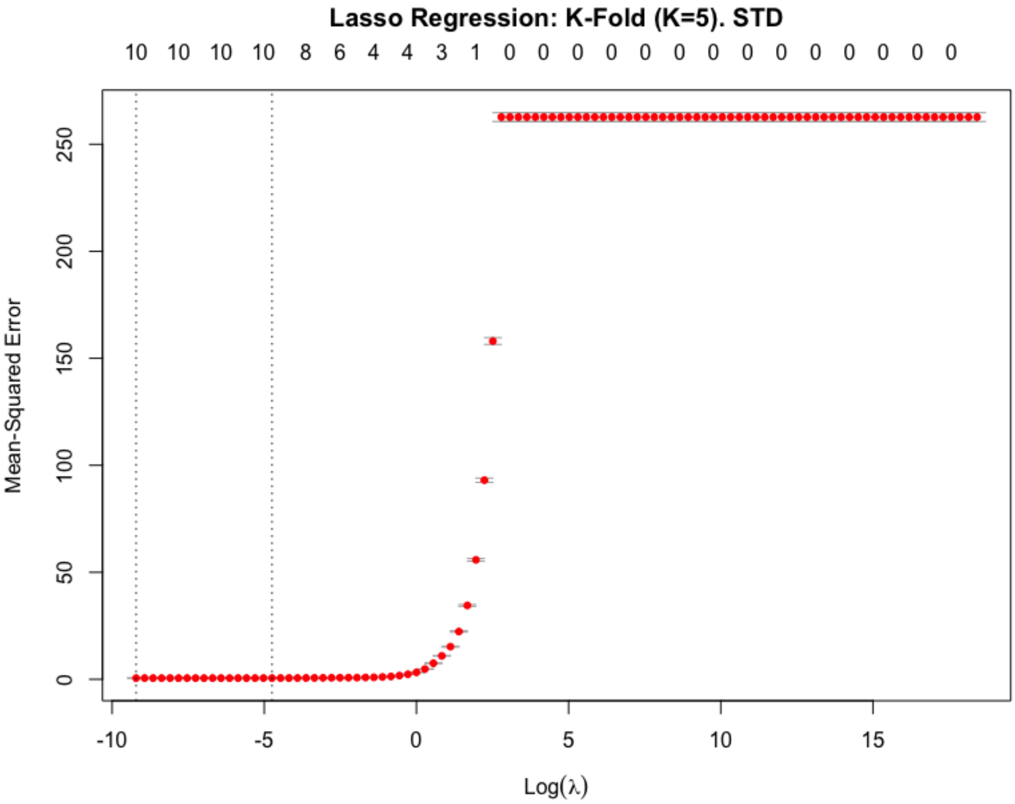
In questo caso i principali risultati ottenuti sono:

1. $\lambda_{\min} = 1e - 04$
2. $\hat{\beta}(\lambda_{\min}) =$

```
 (Intercept)       AT_STD       AP_STD           AH     AFDP_STD     GTEP_STD      TIT_STD
136.454316660 -2.762367391 -0.361258171 -0.008950318 -0.368627944  3.348456485  9.155765880
     TAT_STD      CDP_STD       CO_STD      NOX_STD
-4.541496738  0.971898019  0.049602680 -0.163793650
```

3. $\text{MSE}_{\min} = \mathbf{0.551134}$.

Con il grafico:



5.3 Elastic-Net

La procedura di regolarizzazione Elastic Net consiste nella minimizzazione della seguente funzione di perdita:

$$L_{\text{elastic-net}}(\beta; \lambda_1; \lambda_2) = (Y - X\beta)^{\mathrm{t}}(Y - X\beta) + \lambda_1 * \sum_{j=1}^{k} |\beta_{\mathrm{j}}| + \lambda_2 * \sum_{j=1}^{k} {\beta_{\mathrm{j}}}^2$$

Funzione che può essere formalmente riscritta grazie alla definizione del parametro $\alpha = \frac{\lambda_1}{\lambda_1 + \lambda_2}$.

$$L_{\text{elastic-net}}(\beta; \lambda_1; \lambda_2) = (Y - X\beta)^{\mathrm{t}}(Y - X\beta) + \lambda * \{\alpha \sum_{j=1}^{k} |\beta_{\mathrm{j}}| + (1 - \alpha) * \sum_{j=1}^{k} {\beta_{\mathrm{j}}}^2\}$$

Si può notare come quest'ultima tecnica possa essere considerata come una combinazione tra le prime 2 introdotte. Questo vuol dire che maggiore (minore) sarà il valore del parametro α e maggiore (minore) sarà il contributo che fornisce la tecnica di regolarizzazione LASSO in sfavore (favore) della tecnica Ridge nella stima del modello previsivo.

N.B: sono stati presi in considerazione valori di α pari a 0.2 - 0.4 - 0.6 - 0.8.

5.3.1 Elastic-Net con $\alpha = 0.2$

In questo caso la componente Lasso ha un peso del 20% mentre la componente Ridge un peso del 80% sulla penalità, basta osservare che:

$$L_{\text{elastic-net}}(\beta; \lambda_1; \lambda_2) = (Y - X\beta)^{\mathrm{t}}(Y - X\beta) + \lambda * \{0.20 \sum_{j=1}^{k} |\beta_{\mathrm{j}}| + 0.80 * \sum_{j=1}^{k} {\beta_{\mathrm{j}}}^2\}$$

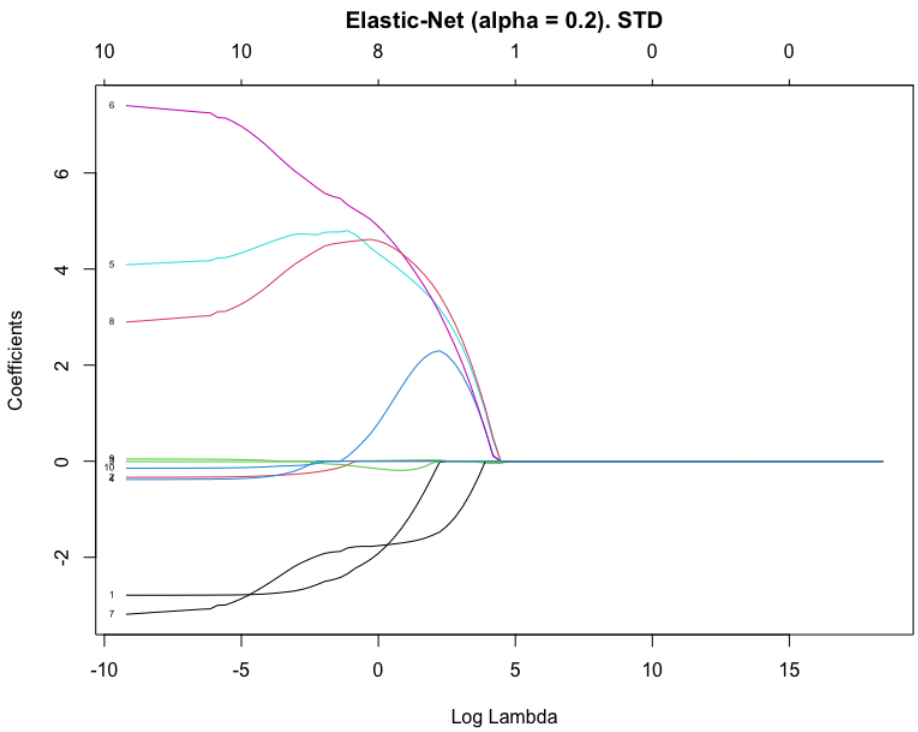


Figura 5: *Si noti la somiglianza con il grafico dei coefficenti visto in precedenza nella figura 3*

K-Fold Cross Validation: K=10

```
listaRes_EN_A02_K10 <- calcola_CV_MSE_Modulare(X_Regressori,Y_Tey,10,0.2)
mse_EN_A02_K10 <- listaRes_EN_A02_K10$mse
lmin_EN_A02_K10 <- listaRes_EN_A02_K10$lam
coefs_EN_A02_K10 <- listaRes_EN_A02_K10$coefficenti
```

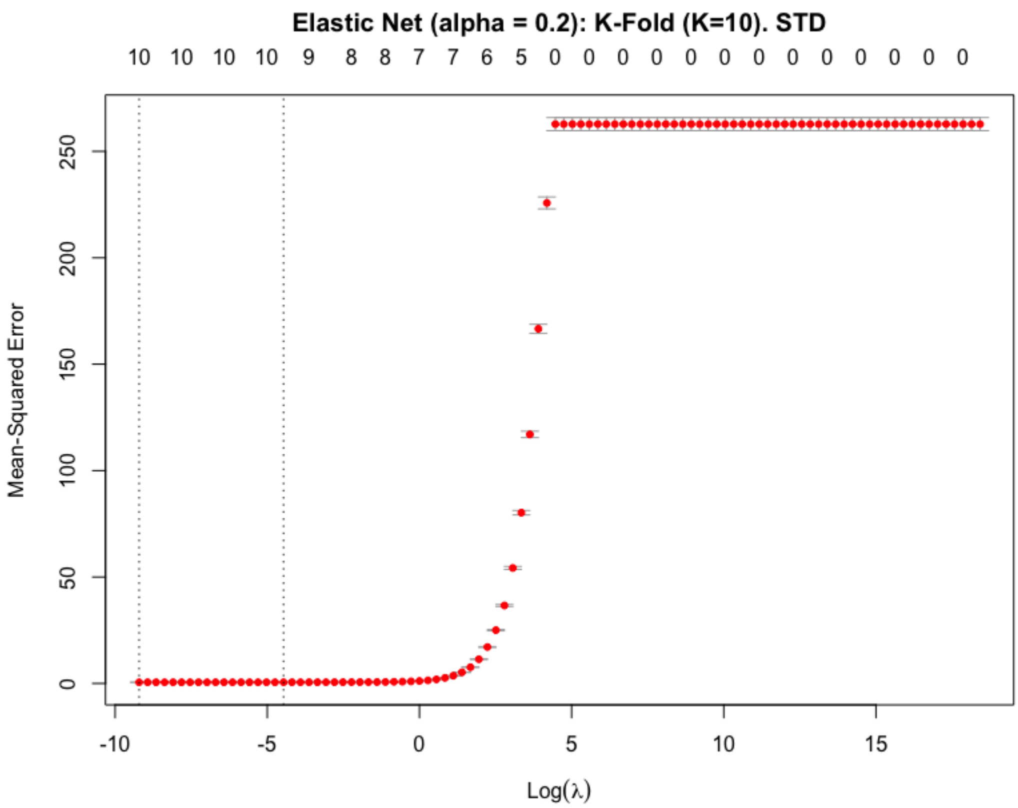
I principali risultati ottenuti sono:

1. $\lambda_{\text{min}} = 1e - 04$
2. $\hat{\beta}(\lambda_{\text{min}}) =$

```
 (Intercept)       AT_STD       AP_STD           AH     AFDP_STD     GTEP_STD      TIT_STD
136.455844654 -2.763370634 -0.361296465 -0.008969617 -0.369656162  3.354590972  9.149495352
     TAT_STD      CDP_STD       CO_STD      NOX_STD
-4.536051478  0.976754019  0.050024681 -0.164096961
```

3. $\text{MSE}_{\text{min}} = \mathbf{0.551398}$.

Con il grafico:



K-Fold Cross Validation: K=5

```
listaRes_EN_A02_K5 <- calcola_CV_MSE_Modulare(X_Regressori,Y_Tey,5,0.2)
mse_EN_A02_K5 <- listaRes_EN_A02_K5$mse
lmin_EN_A02_K5 <- listaRes_EN_A02_K5$lam
coefs_EN_A02_K5 <- listaRes_EN_A02_K5$coefficenti
```

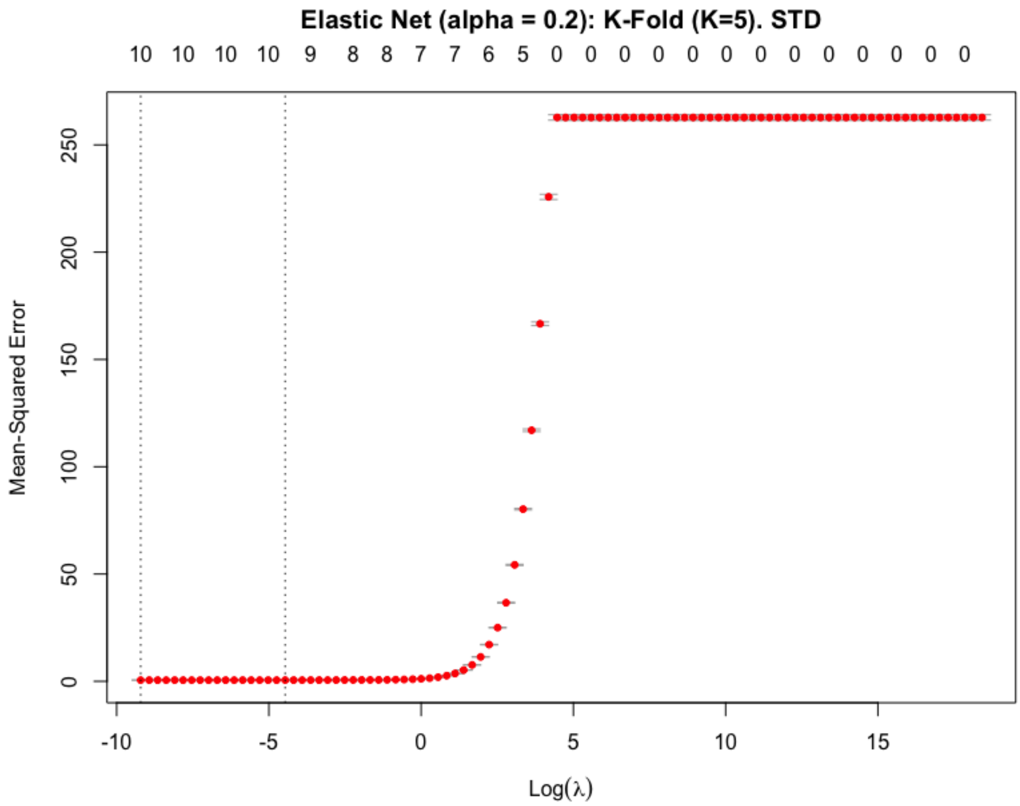
I principali risultati ottenuti sono:

1. $\lambda_{\min} = 1e - 04$
2. $\hat{\beta}(\lambda_{\min}) =$

```
  (Intercept)       AT_STD       AP_STD           AH     AFDP_STD     GTEP_STD      TIT_STD
136.455844654 -2.763370634 -0.361296465 -0.008969617 -0.369656162  3.354590972  9.149495352
      TAT_STD      CDP_STD       CO_STD      NOX_STD
 -4.536051478  0.976754019  0.050024681 -0.164096961
```

3. $\text{MSE}_{\min}$ = **0.552290**.

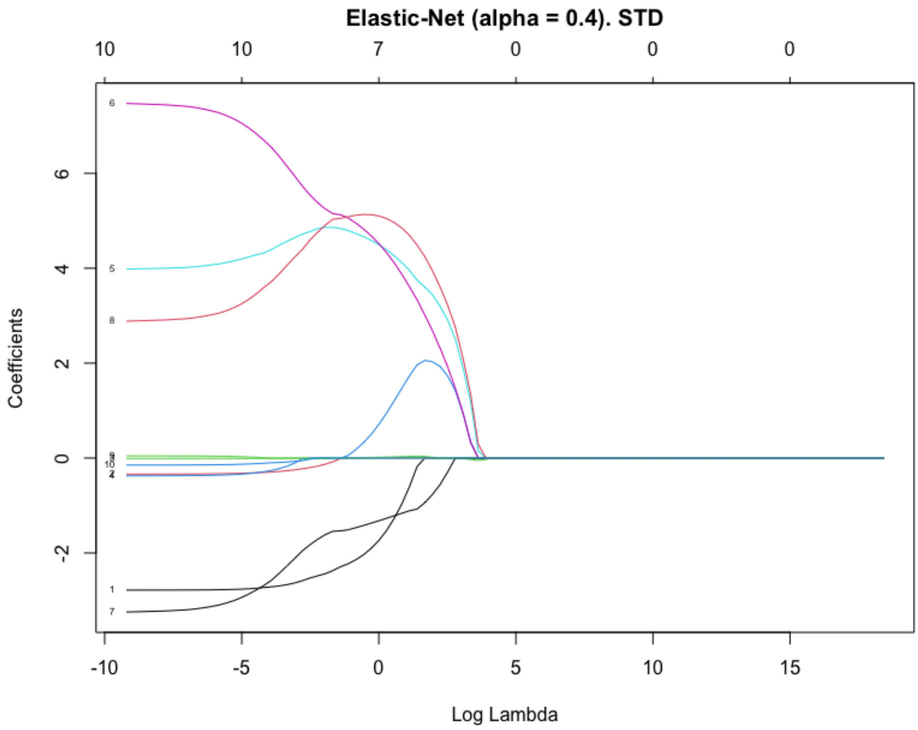
Con il grafico:



5.3.2 Elastic-Net con $\alpha = 0.4$

In questo caso la componente Lasso ha un peso del 40% mentre la componente Ridge un peso del 60% sulla penalità, basta osservare che:

$$L_{\text{elastic-net}}(\beta;\lambda_1;\lambda_2) = (Y - X\beta)^{\text{t}}(Y - X\beta) + \lambda * \{0.40 \sum_{j=1}^{k} |\beta_{\text{j}}| + 0.60 * \sum_{j=1}^{k} {\beta_{\text{j}}}^2\}$$



K-Fold Cross Validation: K=10

```
listaRes_EN_A02_K10 <- calcola_CV_MSE_Modulare(X_Regressori,Y_Tey,10,0.2)
mse_EN_A02_K10 <- listaRes_EN_A02_K10$mse
lmin_EN_A02_K10 <- listaRes_EN_A02_K10$lam
coefs_EN_A02_K10 <- listaRes_EN_A02_K10$coefficenti
```

I principali risultati ottenuti sono:

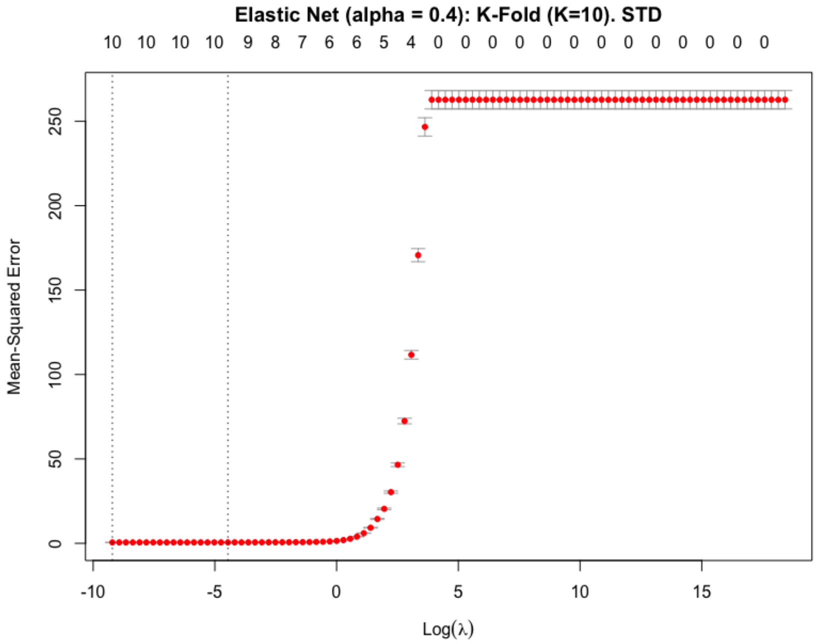
1. $\lambda_{\text{min}} = 1e - 04$

2. $\hat{\beta}(\lambda_{\text{min}}) =$

```
  (Intercept)        AT_STD        AP_STD            AH      AFDP_STD      GTEP_STD       TIT_STD
136.455385782  -2.762980697  -0.361375078  -0.008963821  -0.369338010   3.350504776   9.152307949
      TAT_STD       CDP_STD        CO_STD       NOX_STD
 -4.538521900   0.976043315   0.049904496  -0.164041270
```

3. $\text{MSE}_{\text{min}} = \mathbf{0.550466}$.

Con il grafico:



K-Fold Cross Validation: K=5

```
listaRes_EN_A02_K5 <- calcola_CV_MSE_Modulare(X_Regressori,Y_Tey,5,0.2)
mse_EN_A04_K5 <- listaRes_EN_A04_K5$mse
lmin_EN_A04_K5 <- listaRes_EN_A04_K5$lam
coefs_EN_A04_K5 <- listaRes_EN_A04_K5$coefficenti
```

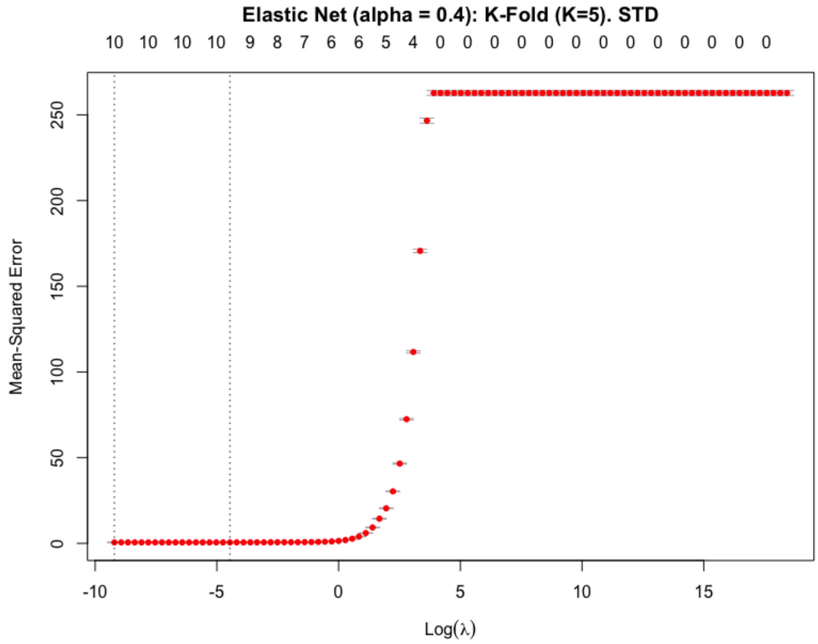
I principali risultati ottenuti sono:

1. $\lambda_{\text{min}} = 1e - 04$
2. $\hat{\beta}(\lambda_{\text{min}}) =$

```
  (Intercept)        AT_STD        AP_STD            AH      AFDP_STD      GTEP_STD       TIT_STD
136.455385782  -2.762980697  -0.361375078  -0.008963821  -0.369338010   3.350504776   9.152307949
      TAT_STD       CDP_STD        CO_STD       NOX_STD
 -4.538521900   0.976043315   0.049904496  -0.164041270
```

3. $\text{MSE}_{\text{min}} =$ **0.550005**.

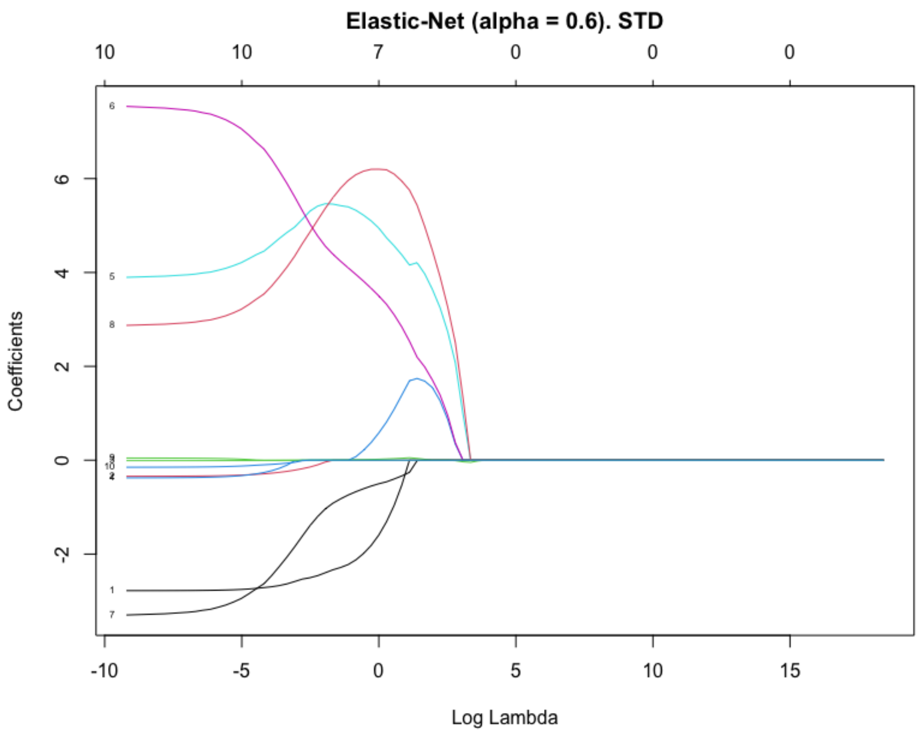
Con il grafico:



5.3.3 Elastic-Net con $\alpha = 0.6$

In questo caso la componente Lasso ha un peso del 60% mentre la componente Ridge un peso del 40% sulla penalità, basta osservare che:

$$L_{\text{elastic-net}}(\beta;\lambda_1;\lambda_2) = (Y - X\beta)^{\text{t}}(Y - X\beta) + \lambda * \{0.60\sum_{j=1}^{k}|\beta_{\text{j}}| + 0.40 * \sum_{j=1}^{k}{\beta_{\text{j}}}^2\}$$



K-Fold Cross Validation: K=10

```
listaRes_EN_A06_K10 <- calcola_CV_MSE_Modulare(X_Regressori,Y_Tey,10,0.2)
mse_EN_A06_K10 <- listaRes_EN_A06_K10$mse
lmin_EN_A06_K10 <- listaRes_EN_A06_K10$lam
coefs_EN_A06_K10 <- listaRes_EN_A06_K10$coefficenti
```

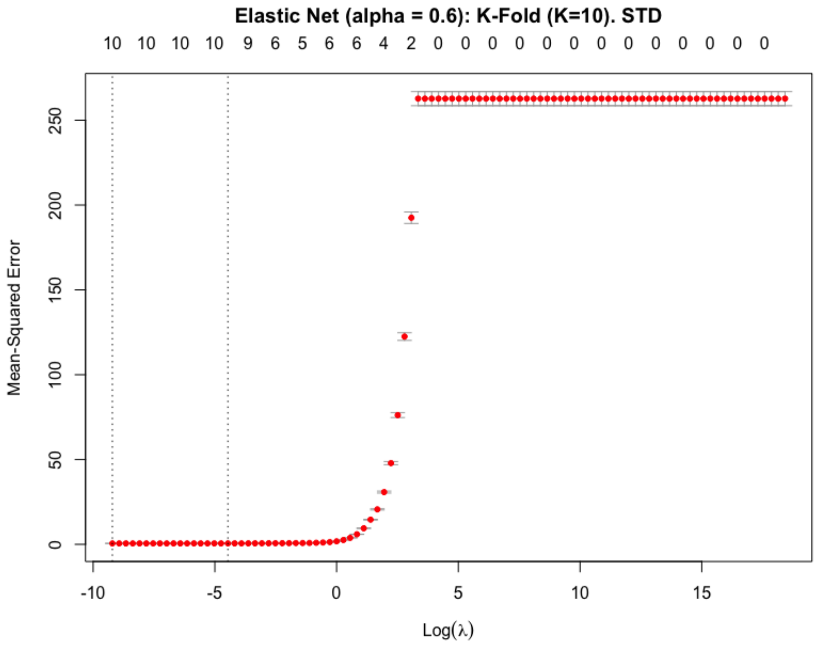
I principali risultati ottenuti sono:

1. $\lambda_{\text{min}} = 1e - 04$
2. $\hat{\beta}(\lambda_{\text{min}}) =$

```
  (Intercept)        AT_STD        AP_STD            AH      AFDP_STD      GTEP_STD       TIT_STD 
136.455080517  -2.762868869  -0.361277395  -0.008959966  -0.369141994   3.351521518   9.152628598 
      TAT_STD       CDP_STD        CO_STD       NOX_STD 
 -4.538772773   0.974331061   0.049813652  -0.163945302 
```

3. $\text{MSE}_{\text{min}} = \mathbf{0.549511}$.

Con il grafico:



K-Fold Cross Validation: K=5

```
listaRes_EN_A06_K5 <- calcola_CV_MSE_Modulare(X_Regressori,Y_Tey,5,0.2)
mse_EN_A06_K5 <- listaRes_EN_A06_K5$mse
lmin_EN_A06_K5 <- listaRes_EN_A06_K5$lam
coefs_EN_A06_K5 <- listaRes_EN_A06_K5$coefficenti
```

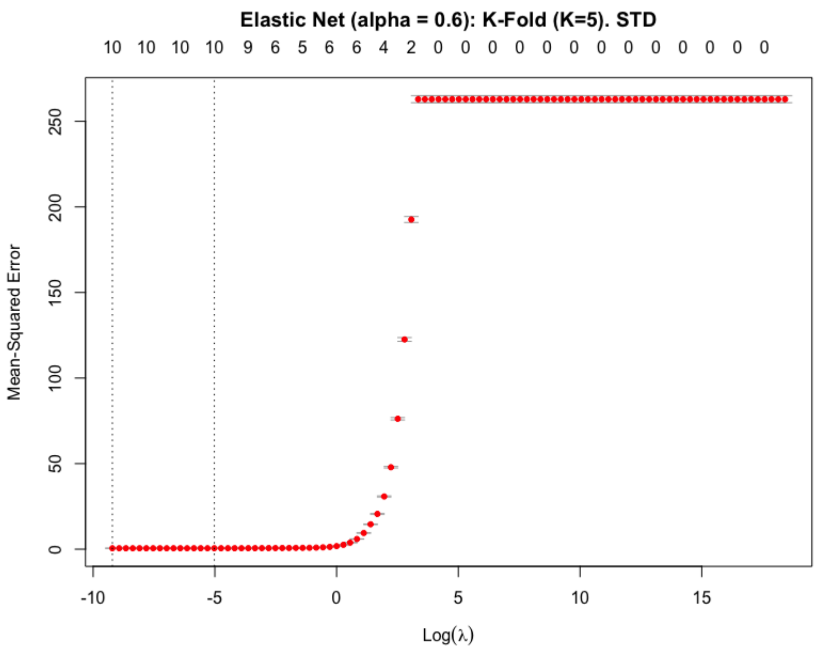
I principali risultati ottenuti sono:

1. $\lambda_{\min} = 1e - 04$
2. $\hat{\beta}(\lambda_{\min}) =$

```
  (Intercept)        AT_STD        AP_STD            AH      AFDP_STD      GTEP_STD       TIT_STD
136.455080517  -2.762868869  -0.361277395  -0.008959966  -0.369141994   3.351521518   9.152628598
      TAT_STD       CDP_STD        CO_STD       NOX_STD
 -4.538772773   0.974331061   0.049813652  -0.163945302
```

3. $\text{MSE}_{\min} = \mathbf{0.549096}$.

Con il grafico:



5.3.4 Elastic-Net con $\alpha = 0.8$

In questo caso la componente Lasso ha un peso del 80% mentre la componente Ridge un peso del 20% sulla penalità, basta osservare che:

$$L_{\text{elastic-net}}(\beta; \lambda_1; \lambda_2) = (Y - X\beta)^{\mathrm{t}}(Y - X\beta) + \lambda * \{0.80 \sum_{j=1}^{k} |\beta_{\mathrm{j}}| + 0.20 * \sum_{j=1}^{k} {\beta_{\mathrm{j}}}^2\}$$

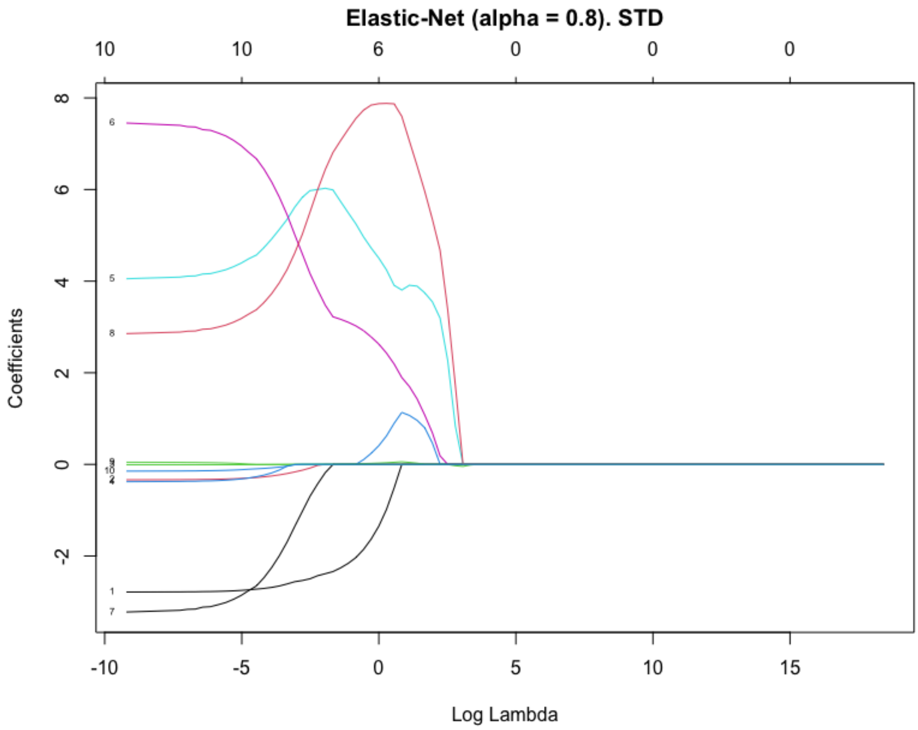


Figura 6: *Si noti la somiglianza con il grafico dei coefficenti visto in precedenza nella figura 4*

K-Fold Cross Validation: K=10

```
listaRes_EN_A08_K10 <- calcola_CV_MSE_Modulare(X_Regressori,Y_Tey,10,0.2)
mse_EN_A08_K10 <- listaRes_EN_A08_K10$mse
lmin_EN_A08_K10 <- listaRes_EN_A08_K10$lam
coefs_EN_A08_K10 <- listaRes_EN_A08_K10$coefficenti
```

I principali risultati ottenuti sono:

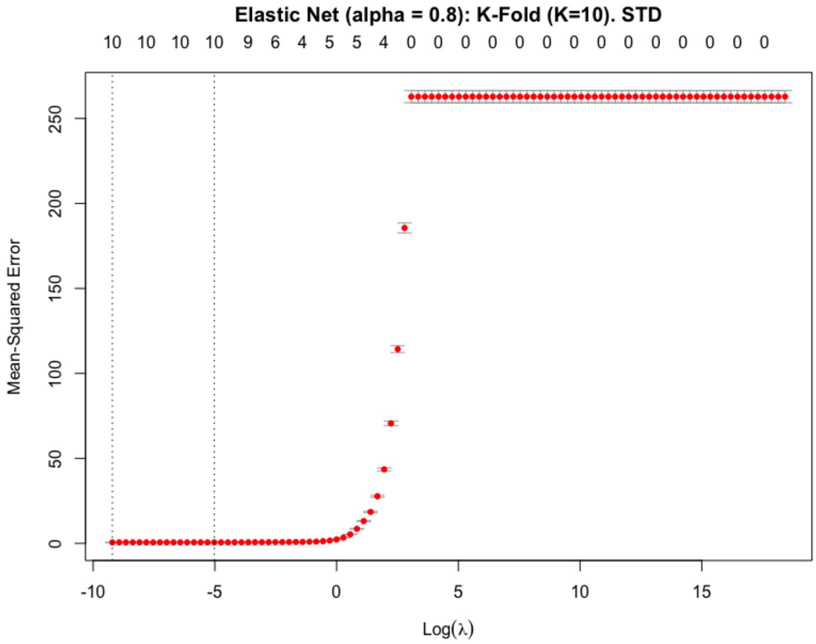
1. $\lambda_{\text{min}} = 1e - 04$

2. $\hat{\beta}(\lambda_{\text{min}}) =$

```
  (Intercept)        AT_STD        AP_STD            AH      AFDP_STD      GTEP_STD       TIT_STD
136.454621850  -2.762479164  -0.361355884  -0.008954172  -0.368823959   3.347438936   9.155443692
      TAT_STD       CDP_STD        CO_STD       NOX_STD
 -4.541244777   0.973613338   0.049693514  -0.163889610
```

3. $\text{MSE}_{\text{min}} =$ **0.549918**.

Con il grafico:



K-Fold Cross Validation: K=5

```
listaRes_EN_A08_K5 <- calcola_CV_MSE_Modulare(X_Regressori,Y_Tey,5,0.2)
mse_EN_A08_K5 <- listaRes_EN_A08_K5$mse
lmin_EN_A08_K5 <- listaRes_EN_A08_K5$lam
coefs_EN_A08_K5 <- listaRes_EN_A08_K5$coefficenti
```

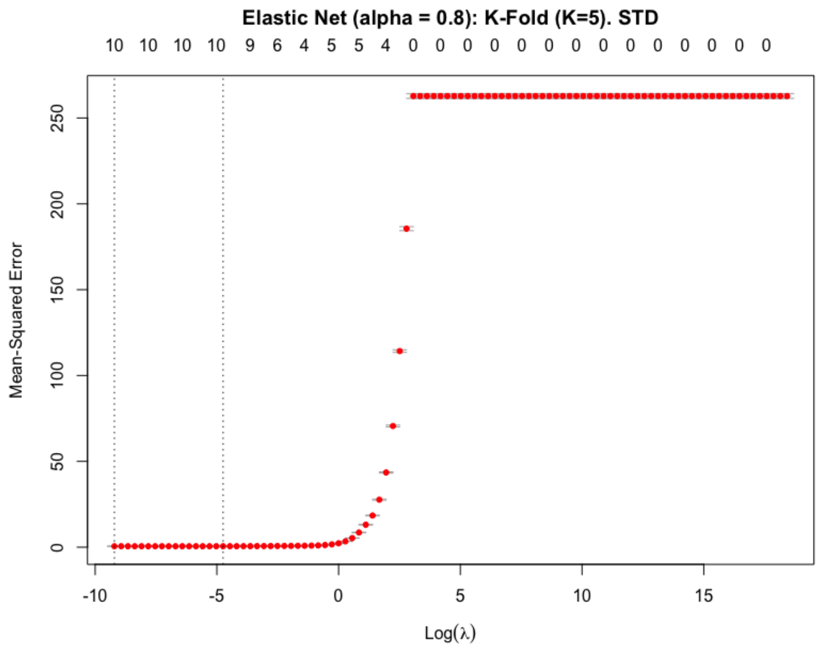
I principali risultati ottenuti sono:

1. $\lambda_{\text{min}} = 1e - 04$
2. $\hat{\beta}(\lambda_{\text{min}}) =$

```
  (Intercept)        AT_STD        AP_STD            AH      AFDP_STD      GTEP_STD       TIT_STD
136.454621850  -2.762479164  -0.361355884  -0.008954172  -0.368823959   3.347438936   9.155443692
      TAT_STD       CDP_STD        CO_STD       NOX_STD
 -4.541244777   0.973613338   0.049693514  -0.163889610
```

3. $\text{MSE}_{\text{min}} = \mathbf{0.550545}$.

Con il grafico:



5.3.5 Confronto e determinazione del migliore modello previsivo

A fronte dei modelli ottenuti nel paragrafo precedente è possibile costruire un vettore la cui struttura è la seguente:

```
     mse_RIDGE_K10 mse_RIDGE_K5 mse_LASSO_K10 mse_LASSO_K5 mse_EN_A02_K10 mse_EN_A02_K5
[1,]     0.5550637    0.5547154     0.5513494    0.5511341       0.551398     0.5522903
     mse_EN_A04_K10 mse_EN_A04_K5 mse_EN_A06_K10 mse_EN_A06_K5 mse_EN_A08_K10 mse_EN_A08_K5
[1,]      0.5504662     0.5500055      0.5495111     0.5490966      0.5499186     0.5505455
```

In cui è possibile osservare che il modello previsivo associato al più piccolo MSE, pari a **0.549096**, è quello caratterizzato dalla tecnica di regolarizzazione Elastic-Net ($\alpha = 0.6$) e tecnica di cross-validation K-Fold con K=5, quest'ultimo restituirà in definitiva le previsioni migliori rispetto ai modelli concorrenti.

Al contrario il peggiore modello previsivo è quello con MSE pari a **0.555063**, raggiunto dal modello successivamente all'applicazione della tecnica Ridge Regression e cross-validation K-Fold con K=10.

5.4 Testing del migliore modello previsivo

In quest'ultima parte della relazione abbiamo focalizzato la nostra attenzione sul processo di valutazione delle performance previsive del modello precedentemente sviluppato. Questa fase di test è stata condotta utilizzando un nuovo insieme di dati raccolti nello stesso contesto, ma provenienti dall'anno successivo rispetto a quello utilizzato per l'addestramento del modello, il 2012.

Per affermare effettivamente che quello ottenuto nel paragrafo precedente sia effettivamente il migliore modello previsivo è necessario confrontare le previsioni sulla variabile dipendente di quest'ultimo con quelle del modello peggiore: se le prime saranno minori delle ultime allora avremo dimostrato la veridicità di quanto ottenuto fin'ora.
Per la fase di testing si è ritenuta nuovamente necessaria una fase di standardizzazione per i regressori in modo tale da poter confrontare correttamente le osservazioni della variabile dipendente con le previsioni ottenute. Il codice è il seguente:

```
file_path2 <- "/Users/francesco/programmi_R/Progetto Finale MS-SL/Datasets/gt_2012.csv"

dati_test <- read.csv(file_path2, header = TRUE, dec = ".", sep = ",")

head(dati_test)

#Bisogna standardizzare i dati in modo da eliminare l'unit  di misura e rendere i dati tra loro comparabili

AT_STD_test <- scale(dati_test$AT)
AP_STD_test <- scale(dati_test$AP)
AH_test <- dati_test$AH #Non c'  bisogno di standardizzare poich  un valore percentuale
AFDP_STD_test <- scale(dati_test$AFDP)
GTEP_STD_test <- scale(dati_test$GTEP)
TIT_STD_test <- scale(dati_test$TIT)
TAT_STD_test <- scale(dati_test$TAT)
TEY_test <- dati_test$TEY
CDP_STD_test <- scale(dati_test$CDP)
CO_STD_test <- scale(dati_test$CO)
NOX_STD_test <- scale(dati_test$NOX)

dati_STD_test <- cbind(AT_STD_test,AP_STD_test,AH_test,AFDP_STD_test,GTEP_STD_test,TIT_STD_test,TAT_STD_test,CDP_STD_test,CO_STD_test,NOX_STD_test)
colnames(dati_STD_test) <- c("AT_STD_test","AP_STD_test","AH_test","AFDP_STD_test","GTEP_STD_test","TIT_STD_test","TAT_STD_test","CDP_STD_test","CO_STD_test","NOX_STD_test")
head(dati_STD_test)
```

A questo punto è possibile calcolare le previsioni della variabile dipendente con il modello che secondo ipotesi restituisce le previsioni con coefficente di bontà della previsione più alto:

```
1 #Previsioni effettuate con il modello ELASTIC-NET alpha = 0.6 e CV K-FOLD (K=5) (migliore assoluto)
2 modello_EN_A06_K5 <- listaRes_EN_A06_K5$modello
3 previsioni_mod_EN_A06_K5 <- predict(modello_EN_A06_K5, newx = dati_STD_test)
4 confronto_osservazioni_previsioni1 <- cbind(TEY_test,previsioni_mod_EN_A06_K5)
5 head(confronto_osservazioni_previsioni)
6 residui_mod_EN_A06_K5 <- TEY_test - previsioni_mod_EN_A06_K5
7 MSE_previsioni_mod_EN_A06_K5 <- mean(residui_mod_EN_A06_K5^2)
8 MSE_previsioni_mod_EN_A06_K5
```

Con il comando a **riga 5** stampiamo una matrice avente n righe e due colonne strutturata come segue:

1. Nella prima colonna sono presenti le osservazioni della variabile dipendente nell'anno 2012;
2. Nella seconda colonna al contrario presiederanno le previsioni della stessa.

```
     TEY_test       s0
[1,]   114.70 119.0558
[2,]   114.72 119.1545
[3,]   114.71 119.2940
[4,]   114.72 119.2674
[5,]   114.72 119.1598
[6,]   114.72 119.0120
```

Con i comandi a **riga 6-7** è stato possibile calcolare l'Errore Quadratico Medio che indica mediamente quanto si discostano le osservazioni della variabile dipendente dalle previsioni. Il risultato è stato **10.52337**.

Effettuando le medesime operazioni in funzione del modello peggiore, ottenuto tramite Ridge Regression e K-Fold (K=10), otteniamo invece un MSE pari a **10.52354**.

Poiché il primo modello presenta un MSE leggermente inferiore possiamo giungere alla conclusione che, in effetti, restituisce le previsioni più accurate sulla variabile dipendente.