Forest Fires

Andrei Petrisor, Antonio Radu, Lorenzo Medici, Andrea Rusconi

2023-12-01

Contents

1 Dataset

Questo dataset è pubblico e a disposizione per la ricerca. I dettagli sul dataset possono essere trovati in Cortez e Morais (2007). Il dataset è composto dalle seguenti variabili:

1. Coordinata spaziale dell'asse X all'interno della mappa del parco Montesinho: da 1 a 9
2. Y coordinata spaziale dell'asse y all'interno della mappa del parco Montesinho: da 2 a 9
3. mese: mese dell'anno: da "gen" a "dic"
4. giorno della settimana: da "lunedì" a "domenica"
5. Indice FFMC dal sistema FWI: da 18,7 a 96,20
6. Indice DMC dal sistema FWI: da 1,1 a 291,3
7. Indice DC dal sistema FWI: da 7,9 a 860,6
8. Indice ISI del sistema FWI: da 0,0 a 56,10
9. temperatura temporanea in gradi Celsius: da 2,2 a 33,30
10. Umidità relativa RH in %: da 15,0 a 100
11. velocità del vento in km/h: da 0,40 a 9,40
12. pioggia in mm/m2: da 0,0 a 6,4
13. area della superficie bruciata della foresta (in ettari): da 0,00 a 1090,84.

1.1 Obiettivo

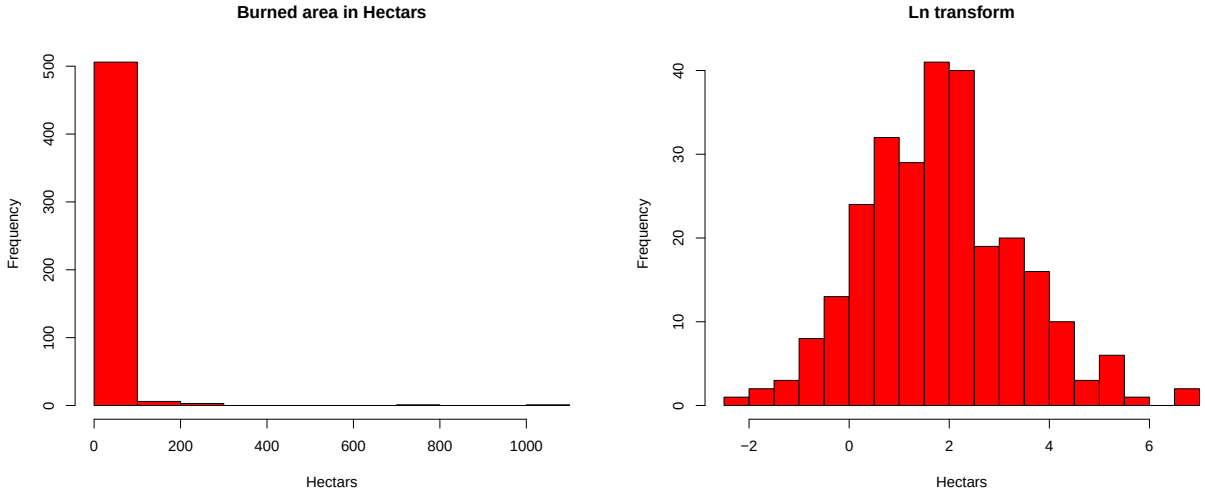
In questo dataset siamo interessati a modellare l'area bruciata della foresta come funzione delle altre variabili.

Cortez P. e Morais A. "Un approccio di data mining per prevedere gli incendi boschivi utilizzando dati meteorologici." In J. Neves, MF Santos e J. Machado Eds., "Nuove tendenze nell'intelligenza artificiale", Atti della 13a EPIA 2007 Conferenza portoghese sull'intelligenza artificiale, dicembre, Guimaraes, Por- tugal, pp. 512-523, 2007. APPIA, ISBN-13 978-989-95618-0-9. 18-0-9. Disponibile a: http://www3.dsi.uminho.pt/pcortez/fires.pdf

Il dataset è composto dalle seguenti rilevazioni:

```
##       Days
## Months mon tue wed thu fri sat sun
##    jan   0   0   0   0   0   1   1
##    feb   3   2   1   1   5   4   4
##    mar  12   5   4   5  11  10   7
##    apr   1   0   1   2   1   1   3
##    may   0   0   0   0   1   1   0
##    jun   3   0   3   2   3   2   4
##    jul   4   6   3   3   3   8   5
##    aug  15  28  25  26  21  29  40
##    sep  28  19  14  21  38  25  27
##    oct   4   2   2   0   1   3   3
##    nov   0   1   0   0   0   0   0
##    dic   0   0   0   0   0   0   0
```

1.2 Histogrammi area bruciata



Come possiamo vedere dall'istogramma(sin) i dati sono assimetrici verso lo zero, possiamo percio valutare i dati facendo la trasformata grazie al logaritmo, cosifacendo otteniamo un grafico più preciso.

```
summary(forest$area)
```

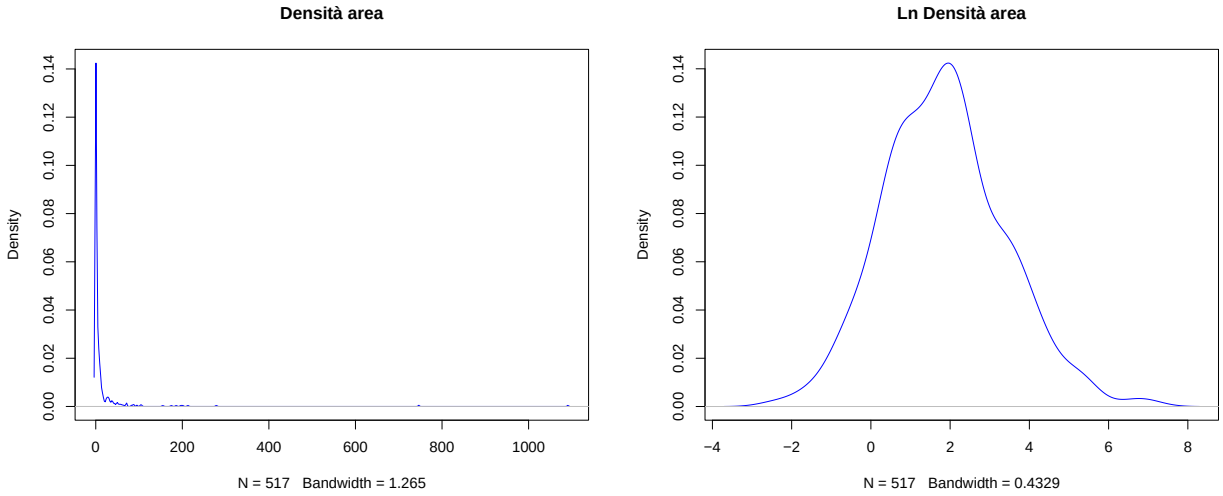
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.52   12.85    6.57 1090.84
```

Facendo il summary otteniamo i valori di: min, max, media, mediana, possiamo considerare i nostri dati molto vicini allo zero.

```
## [1] 0.4777563
```

Si evidenzia che il dataset ha il 48% dei valori che valgono 0.

1.3 Grafici densità area bruciata



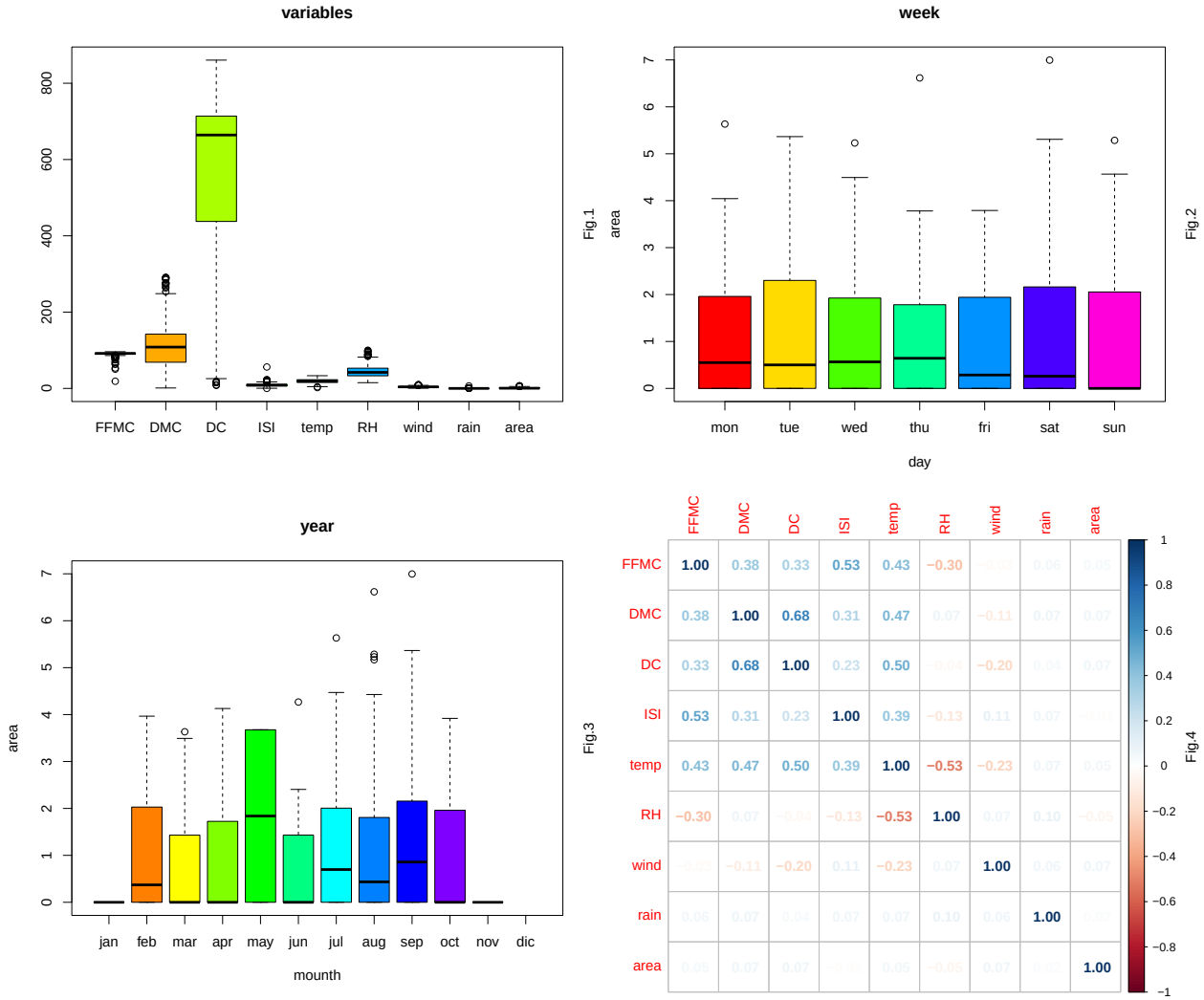
Dobbiamo tenere in considerazione del numero di zeri presenti per considerare un andamento normale, visto

che log(0)= -inf

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.4187  1.1110  2.0242  6.9956
```

come possiamo vedere la mediana e la media sono diminuite.

```
## Warning: il pacchetto 'corrplot' è stato creato con R versione 4.3.2
```



-Come possiamo vedere da questo correlation plot(fig.4) le variabili che sono correlate positivamente di più sono: DC con DMC, ISI e FFMC, temp e DC. Le variabili correlate negativamente di più sono, invece: RH e temp.

-Dal box plot delle variabili(fig.1) possiamo determinare che le mediane di quasi tutte le variabili sono più o meno simili quindi ci sono poche differenze,inoltre possiamo notare come in tutte le variabili ci siano degli outliers sia al minimo che al massimo.La variabile DMC,invece,presenta dei baffi più lunghi il che implica che tale variabile ha valori più incoerenti rispetto alle altre, la mediane lontana da tutte le altre e spostata molto verso il terzo quartile.

-Dal box plot dei giorni della settimana(fig.2) possiamo vedere come le mediane sono simili tra di loro come anche i baffi, questo dimostra come i giorni della settimana hanno tutti valori coerenti tra di loro.

-Dal box plot dei mesi dell'anno(fig.3) notiamo come il mese di dicembre e quello di maggio siano quelli più incoerenti rispetto agli altri mesi. Il mese di agosto presenta molti più outliers rispetto agli altri mesi.

2 Testing

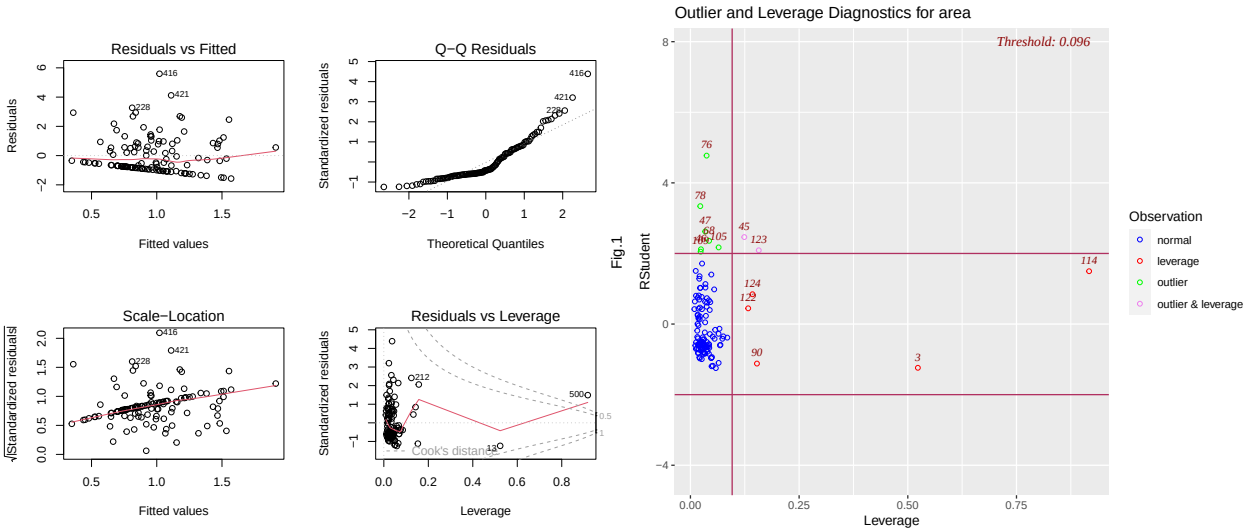
Per la fase di testing prendiamo in considerazione solamente il mese di agosto, per poter spiegare come varia l'area bruciata in relazione alle altre variabili. Dividiamo la fase di Testing in: train e test, con il train al 70% e il test al 30%. Questo è il nostro nuovo dataset:

Proviamo a fare un modello lineare con i dati dai mesi di agosto.

```
##
## Call:
## lm(formula = area ~ rain + RH + temp + FFMC + DMC, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5702 -0.8345 -0.5287  0.6288  5.5955
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.292937   3.540255  -0.365   0.7156
## rain         0.044054   0.212911   0.207   0.8364
## RH           0.014859   0.012451   1.193   0.2351
## temp         0.083263   0.039326   2.117   0.0363 *
## FFMC         0.001123   0.035157   0.032   0.9746
## DMC         -0.002174   0.002545  -0.854   0.3947
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.3 on 119 degrees of freedom
## Multiple R-squared:  0.04681,    Adjusted R-squared:  0.006758
## F-statistic: 1.169 on 5 and 119 DF,  p-value: 0.3286
```

Come possiamo vedere il nostro modello non performa benissimo, con un 4% come r^2 il modello ha un grande margine di errore. Tra le varie covariate scelte notiamo come la temperatura abbia il valore piu grande, all'aumentare unitario dell'area, la temperatura media aumenta di 0.08. Proviamo a vedere se ci sono degli Outliers(valori estremi) nel modello.

```
## Warning: il pacchetto 'olsrr' è stato creato con R versione 4.3.2
```



Possiamo vedere come nel grafici (Fig.1), un numero considerevole di osservazioni non segue l'andamento desiderato, poniamo maggiore attenzione sul grafico Q-Q Res. dove possiamo vedere che i vari valori alle code tendono a spostarsi di molto, l'andamento non tende per niente alla normale. Per quanto riguarda il grafico a destra possiamo notare come alcune rilevazioni siano parecchio rispetto alle altre, poniamo attenzione soprattuto a quelle che sono Outlier&Leverage e le rimuoviamo.

```
##
## Call:
## lm(formula = area ~ rain + RH + temp + FFMC + DMC, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5217 -0.6957 -0.3932  0.7101  3.0223
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.800705   3.873662   0.465 0.642927
## rain        -0.876780   0.568884  -1.541 0.126057
## RH           0.025039   0.010737   2.332 0.021466 *
## temp         0.115512   0.033134   3.486 0.000699 ***
## FFMC        -0.043793   0.040497  -1.081 0.281822
## DMC         -0.003424   0.002091  -1.638 0.104255
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.049 on 113 degrees of freedom
## Multiple R-squared:  0.1131, Adjusted R-squared:  0.07389
## F-statistic: 2.883 on 5 and 113 DF,  p-value: 0.01732
```

Possiamo vedere un leggero miglioramento anche se non di tanto, possiamo vedere come l'r^2 sia aumentato del 7% rendendo il modello un minimo piu preciso. Notiamo anche come la covariata piu significativa sia la temperatura , quindi all'aumentare unitario dell'area, la temperatura media aumenta di 0.12. I coefficienti associati alle variabili pioggia, FFMC, DMC non sono significativi, quindi supponiamo non influiscano sull'area, passando non avere la pioggia ad averla, a parità di temperatura , l'area diminuisce di 0.87.

2.1 Confronto tra modelli:

```
## AIC lm 3 BIC lm 3 AIC lm 2 BIC lm 2 AIC lm 1 BIC lm 1
## 369.1109 413.5769 356.9427 376.3966 428.1296 447.9278
```

2.2 Previsioni con Model2:

```
#use lasso regression model to predict response value
new = test
previsioni_mod3 = predict(model3, newdata = new)#previsione puntuale

#find SST and SSE
sst <- sum((test$area - mean(test$area))^2)
sse <- sum((previsioni_mod3 - test$area)^2)

# Root Mean Squared Error: è una misura dell'errore che compiamo
sqrt(mean((test$area - previsioni_mod3)^2))#--
```

```
## [1] 1.59681
```

```
# intervallo di previsione
predict(model3,new,interval="predict", level=0.95)#intervallo di previsione
```

```
##           fit         lwr      upr
## 6   0.5836614 -1.66733787 2.834661
## 24  0.6166137 -1.64652253 2.879750
## 25  0.7278954 -1.65179411 3.107585
## 54  0.5815156 -1.63947229 2.802504
## 74  0.6324137 -1.66650183 2.931329
## 80  0.2724269 -2.13804265 2.682896
## 82  0.8101871 -1.39039184 3.010766
## 93  0.4255444 -1.83756287 2.688652
## 100 0.4039733 -1.79103071 2.598977
## 102 0.8606814 -1.41250872 3.133871
## 108 0.8746194 -1.32119707 3.070436
## 136 0.5252190 -1.84690164 2.897340
## 143 0.8686580 -1.51356234 3.250878
## 146 0.9610996 -1.24180041 3.164000
## 159 0.4574649 -1.75671085 2.671641
## 176 0.8659322 -1.36376069 3.095625
## 180 0.8606814 -1.41250872 3.133871
## 196 0.7404783 -1.52244291 3.003400
## 207 0.4944697 -1.91266388 2.901603
## 230 0.3928708 -1.89220211 2.677944
## 236 0.4070499 -1.78827624 2.602376
## 243 0.6958739 -1.51988298 2.911631
## 245 0.9658094 -1.26570555 3.197324
## 246 1.1612504 -1.05195404 3.374455
## 248 1.5526706 -0.78762401 3.892965
## 250 0.7834953 -1.44042093 3.007412
## 251 0.8059086 -1.38799021 2.999807
```

```
## 256 1.7489365 -0.52889292 4.026766
## 259 1.4374189 -0.87383509 3.748673
## 260 1.2192407 -1.05621939 3.494701
## 262 0.5519508 -1.78697747 2.890879
## 263 0.9713831 -1.27425360 3.217020
## 264 0.5020134 -1.74979673 2.753824
## 271 1.5952949 -0.61535818 3.805948
## 272 1.3333886 -0.86083984 3.527617
## 374 0.4483114 -1.77500508 2.671628
## 377 0.4391679 -1.79311938 2.671455
## 378 0.5845791 -1.63558114 2.804739
## 385 0.3942483 -1.86463943 2.653136
## 389 0.4136975 -1.84020603 2.667601
## 403 0.6236329 -1.61431718 2.861583
## 414 0.7984662 -1.42343994 3.020372
## 420 0.8323344 -1.35552148 3.020190
## 426 0.9163905 -1.29836097 3.131142
## 428 0.7025606 -1.54585168 2.950973
## 439 0.9892179 -1.22773129 3.206167
## 442 0.3086896 -1.94108114 2.558460
## 452 0.7226625 -1.70190382 3.147229
## 455 0.3897783 -1.82123207 2.600789
## 458 0.3831688 -1.86657242 2.632910
## 460 0.2062347 -2.13183501 2.544304
## 485 1.7089481 -0.52677025 3.944666
## 491 1.1697498 -1.04110254 3.380602
## 499 1.8957461 -0.37251539 4.164008
## 503 0.9929685 -1.25572317 3.241660
## 505 1.1710071 -1.07828823 3.420302
## 506 1.6411585 -0.61976494 3.902082
## 509 1.4165829 -0.82567785 3.658844
## 512 2.4304251  0.04198607 4.818864
```

2.3 LASSO regression:

```
#penalizza le covariate,avra una parte classica piu un errore
# install.packages("glmnet") # se non è già stato installato
library(glmnet)
```

```
## Warning: il pacchetto 'glmnet' è stato creato con R versione 4.3.2
```

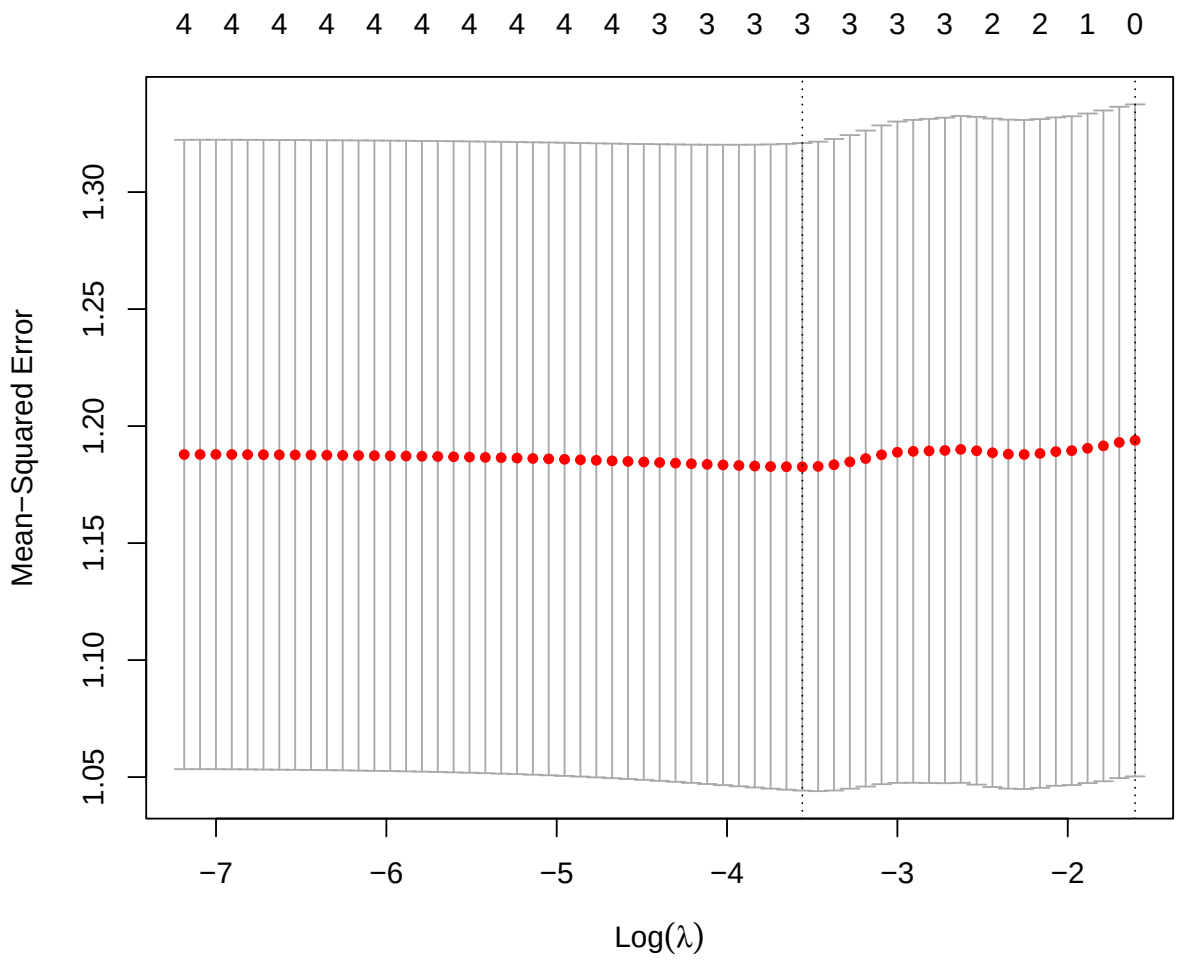
```
## Warning: il pacchetto 'Matrix' è stato creato con R versione 4.3.2
```

```
#define response variable
y <- train$area

#define matrix of predictor variables (uso solo poche variabili ma potete farlo con tutte da togliere p
x <- data.matrix(train[, c("wind","rain","temp","FFMC")])

#perform k-fold cross-validation to find optimal lambda value, la cross-validation è un ottimo modo per
cv_model <- cv.glmnet(x, y, alpha = 1)
```

```
#find optimal lambda value that minimizes test MSE
best_lambda <- cv_model$lambda.min
best_lambda#miglior lambda per penalizzare
```

```
## [1] 0.02848813
```

```
#produce plot of test MSE by lambda value
plot(cv_model)
```




```
# Fittiamo il modello con il best lambda (penalizzazione)

best_model <- glmnet(x, y, alpha = 1, lambda = best_lambda)
coef(best_model)
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                      s0
## (Intercept)  4.76362952
```

```
## wind          .
## rain       -0.22686231
## temp        0.04715436
## FFMC       -0.05336495
```

2.4 Previsioni con Lasso:

```r
#use lasso regression model to predict response value
new = data.matrix(test[,c("wind","rain","temp","FFMC")])
previsioni = predict(best_model, s = best_lambda, newx = new)

# Root Mean Squared Error (RMSE): è una misura dell'errore che compiamo
sqrt(mean((test$area - previsioni)^2))#errore minimo
```

```
## [1] 1.525233
```

Come ppossiamo vedere le previsioni con la Lasso sono migliori rispetto a .....(non so come si chiama l'altro)