

Forest Fires

Andrei Petrisor, Antonio Radu, Lorenzo Medici, Andrea Rusconi

2023-12-01

Contents

1	Dataset	2
1.1	Obiettivo	2
1.2	Histogrammi area bruciata	3
1.3	Grafici densità area bruciata	3
2	Testing	5
2.1	Confronto tra modelli:	10
2.2	Previsioni con Model2:	10
2.3	LASSO regression:	14
2.4	Previsioni con Lasso:	16
2.5	Modello con soglia scelta: logit - glm	16

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

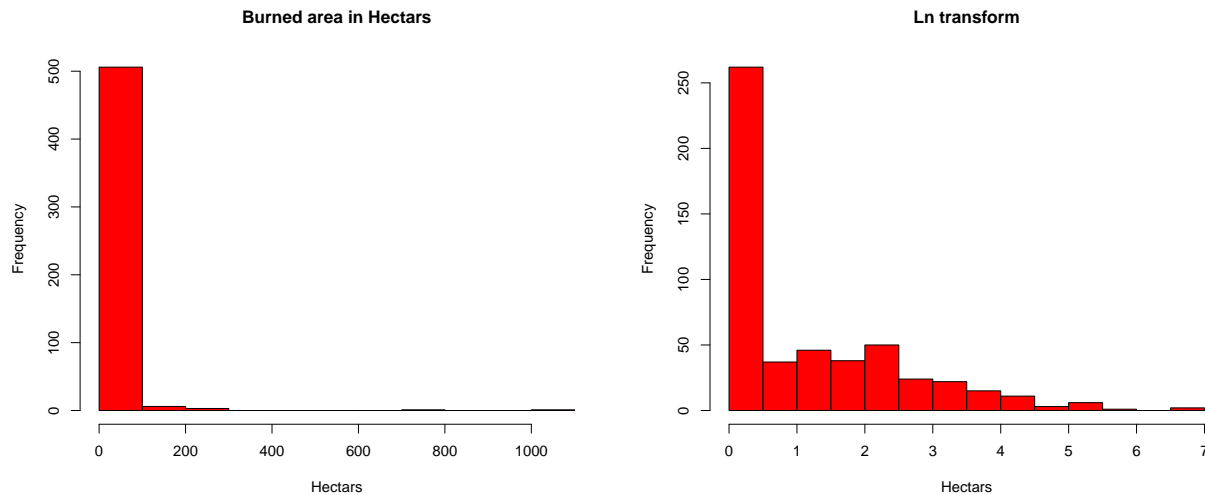
In questo dataset siamo interessati a modellare l'area bruciata della foresta come funzione delle altre variabili.

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, Portugal, 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 asimmetrici verso lo zero, possiamo perciò valutare i dati facendo la trasformata grazie al logaritmo, così facendo otteniamo un grafico più preciso.

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

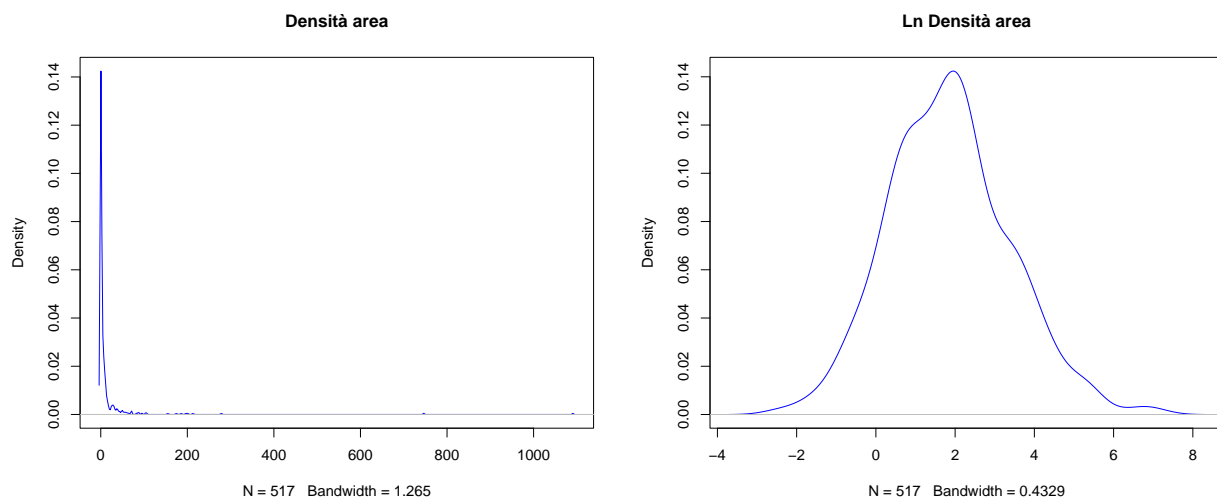
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.00   0.00   0.52   12.85   6.57 1090.84
```

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



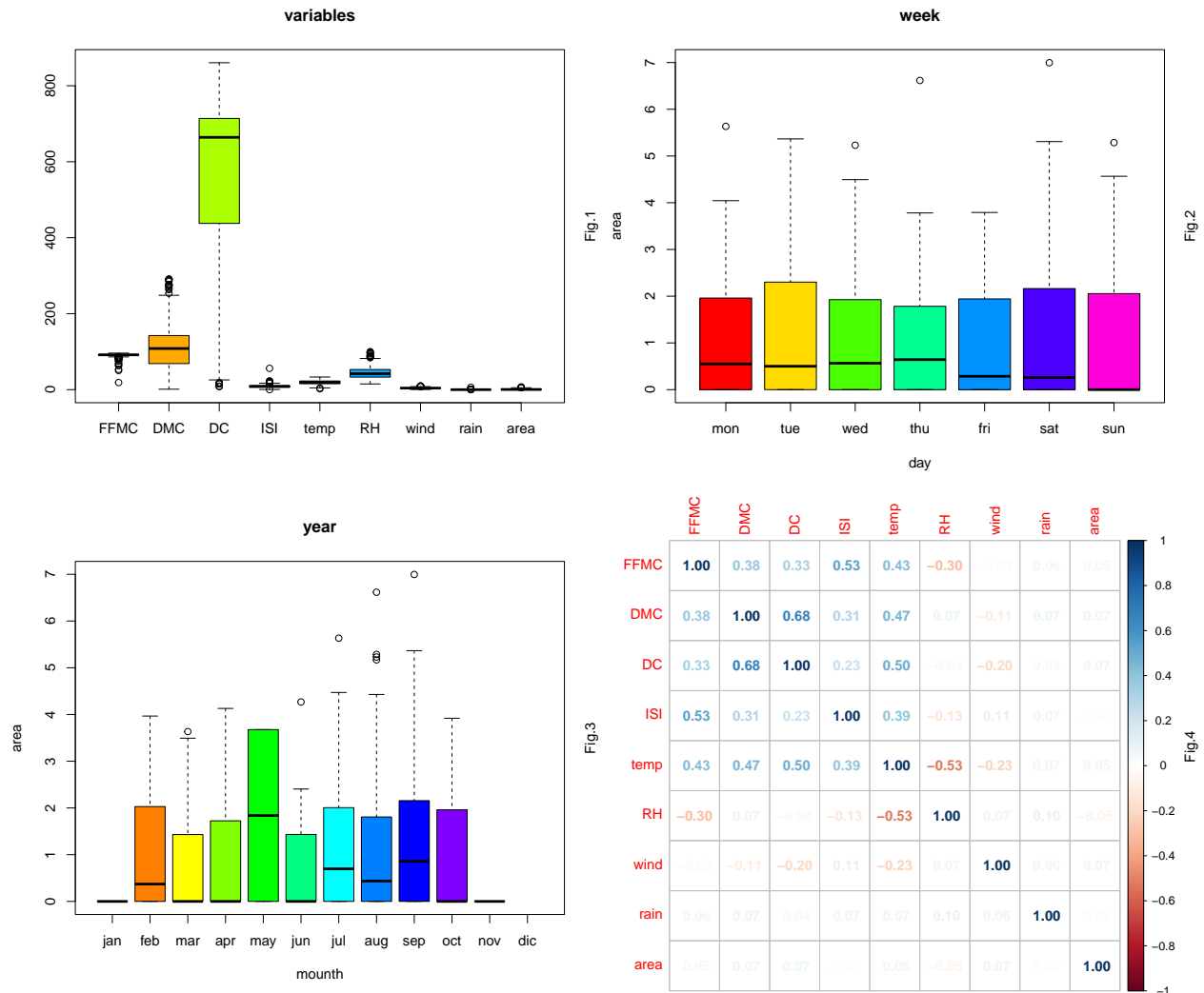
Dobbiamo tenere in considerazione del numero di zeri presenti per considerare un andamento normale, visto

che $\log(0) = -\text{inf}$

```
##      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 mediana 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

```
# si divide il dataset in train e test (validation) così che si fitta il modello sul train e si guarda

# Optional:
forest <- fastDummies::dummy_cols(forest, remove_first_dummy = TRUE)[-c(1,2)]

# Regole classiche sono 70% training e 30% test (o 80-20 a vostra scelta)
set.seed(125) # il seme serve per riprodurre le analisi (reproducibilità del codice)
sample <- sample(c(TRUE, FALSE), nrow(forest), replace=TRUE, prob=c(0.7,0.3))
train  <- forest[sample, ]
test   <- forest[!sample, ]

# Modello con una sola variabile:
model = lm(area ~ rain, data = train)

summary(model)
```

```
##
## Call:
## lm(formula = area ~ rain, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.1169 -1.1169 -0.6982  0.9086  5.8787
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.11694    0.07539   14.82  <2e-16 ***
## rain        -1.41984    0.92201   -1.54    0.124
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.409 on 351 degrees of freedom
## Multiple R-squared:  0.006711, Adjusted R-squared:  0.003881
## F-statistic: 2.371 on 1 and 351 DF, p-value: 0.1245
```

```
# Modello con tutte le variabili:
modell1 = lm(area ~ ., data = train)
summary(modell1)
```

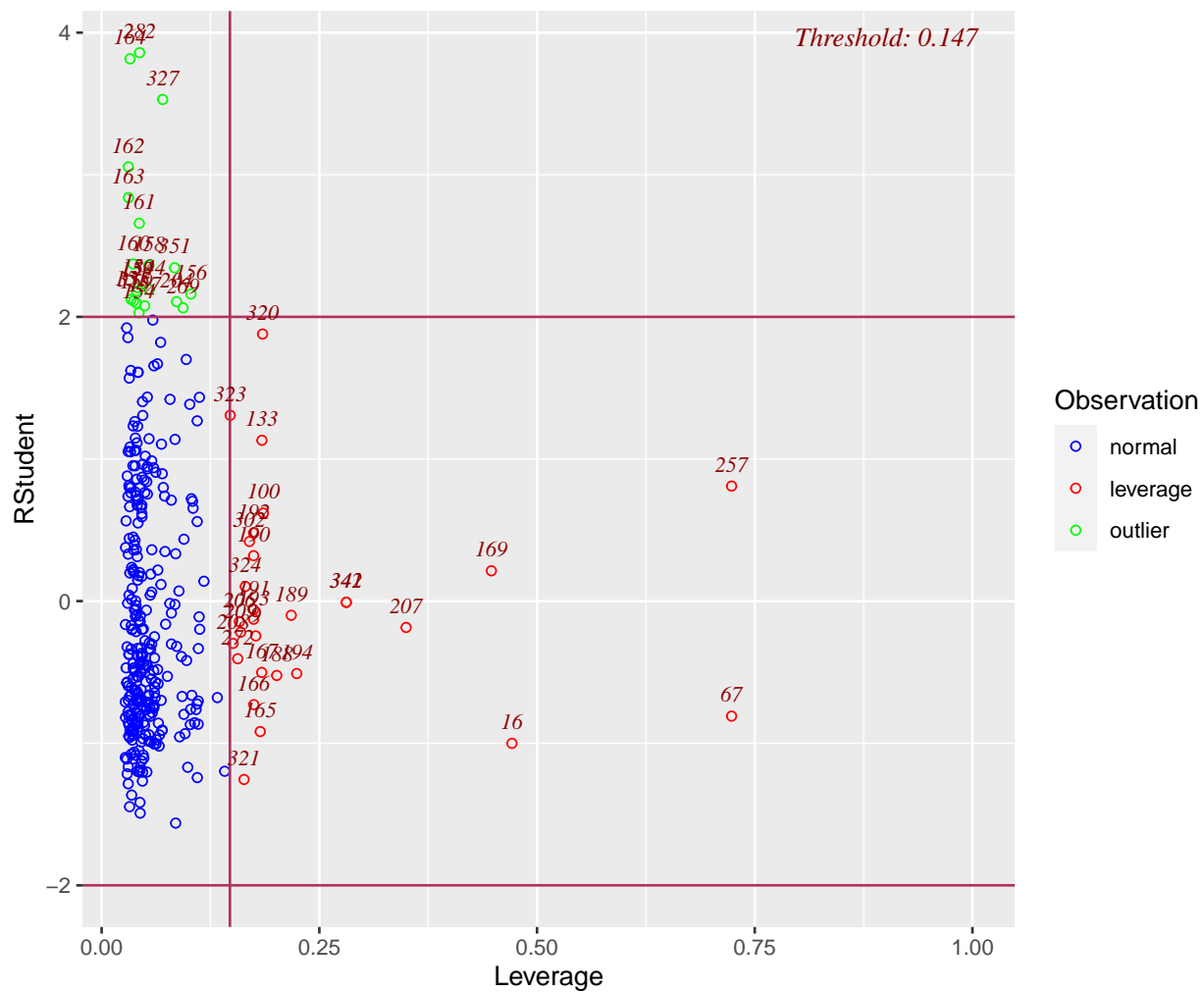
```
##
## Call:
## lm(formula = area ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.0509 -0.9930 -0.4196 0.9067 5.0839
##
## Coefficients: (1 not defined because of singularities)
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.104088  2.062474  -0.535  0.5928
## FFMC        0.007967  0.021031   0.379  0.7051
## DMC         0.003311  0.002244   1.476  0.1410
## DC         -0.002105  0.001694  -1.242  0.2150
## ISI         0.001891  0.020749   0.091  0.9275
## temp        0.044480  0.026364   1.687  0.0925 .
## RH          0.005247  0.007595   0.691  0.4901
## wind        0.102497  0.045852   2.235  0.0261 *
## rain       -1.781002  0.955096  -1.865  0.0631 .
## month_aug   0.248307  1.070361   0.232  0.8167
## month_dec   2.213667  0.994994   2.225  0.0268 *
## month_feb   0.408489  0.650106   0.628  0.5302
## month_jan  -0.541694  1.307862  -0.414  0.6790
## month_jul  -0.037427  0.896186  -0.042  0.9667
## month_jun  -0.696132  0.824498  -0.844  0.3991
## month_mar  -0.379072  0.576235  -0.658  0.5111
## month_may   2.705751  1.487261   1.819  0.0698 .
## month_nov      NA         NA         NA         NA
## month_oct   1.211886  1.261942   0.960  0.3376
## month_sep   1.021802  1.208243   0.846  0.3983
## day_mon     0.207708  0.274390   0.757  0.4496
## day_sat     0.616573  0.269198   2.290  0.0226 *
## day_sun     0.419913  0.263989   1.591  0.1127
## day_thu     0.473808  0.300514   1.577  0.1158
## day_tue     0.506912  0.296819   1.708  0.0886 .
## day_wed     0.121176  0.297501   0.407  0.6840
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.376 on 328 degrees of freedom
## Multiple R-squared:  0.1151, Adjusted R-squared:  0.05033
## F-statistic: 1.777 on 24 and 328 DF, p-value: 0.01496
```

```
# Diagnostic plot del modello:
par(mfrow=c(2,2)) # finestra grafica 2x2
plot(model1)
```

```
## Warning: non si riesce a fare il plot senza sfruttarne uno:
##      322
```


Outlier and Leverage Diagnostics for area



```
# Removing observations guardando numero dell'osservazione:
train = train[-c(480,380,105),]

# Rimuovere osservazioni basandoci sui valori delle variabili:
# train=train[!(train$temp>300),]

# Modello 1 senza outlier:
model2 = lm(area ~ ., data = train)
summary(model2)
```

```
##
## Call:
## lm(formula = area ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0546 -0.9958 -0.4130  0.9061  5.0801
##
## Coefficients: (1 not defined because of singularities)
```



```
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.138314   2.064694  -0.551  0.5818
## FPMC        0.008022   0.021048   0.381  0.7034
## DMC         0.003353   0.002247   1.492  0.1365
## DC          -0.002139   0.001696  -1.261  0.2083
## ISI         0.001644   0.020768   0.079  0.9369
## temp        0.046366   0.026524   1.748  0.0814 .
## RH          0.005453   0.007606   0.717  0.4739
## wind        0.101662   0.045904   2.215  0.0275 *
## rain       -1.779536   0.955853  -1.862  0.0635 .
## month_aug   0.245346   1.071216   0.229  0.8190
## month_dec   2.245021   0.996804   2.252  0.0250 *
## month_feb   0.412339   0.650644   0.634  0.5267
## month_jan  -0.542138   1.308897  -0.414  0.6790
## month_jul  -0.047500   0.897013  -0.053  0.9578
## month_jun  -0.703500   0.825219  -0.853  0.3946
## month_mar  -0.379024   0.576691  -0.657  0.5115
## month_may   2.699104   1.488468   1.813  0.0707 .
## month_nov   NA         NA         NA     NA
## month_oct   1.223977   1.263060   0.969  0.3332
## month_sep   1.035993   1.209372   0.857  0.3923
## day_mon     0.209033   0.274614   0.761  0.4471
## day_sat     0.632568   0.270394   2.339  0.0199 *
## day_sun     0.421203   0.264204   1.594  0.1118
## day_thu     0.474003   0.300752   1.576  0.1160
## day_tue     0.507397   0.297055   1.708  0.0886 .
## day_wed     0.121490   0.297736   0.408  0.6835
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.377 on 327 degrees of freedom
## Multiple R-squared:  0.1164, Adjusted R-squared:  0.0515
## F-statistic: 1.794 on 24 and 327 DF,  p-value: 0.01363
```

```
# Modello con variabili scelte da noi (scelte casualmente al momento):
model3 = lm(area ~ month_dec+wind+rain+temp+I(FFMC*month_aug), data = train)
summary(model3)
```

```
##
## Call:
## lm(formula = area ~ month_dec + wind + rain + temp + I(FFMC *
##     month_aug), data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.6371 -1.0587 -0.6195  0.8999  5.6030
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.225277   0.333634   0.675  0.5000
## month_dec      1.535108   0.625141   2.456  0.0146 *
## wind           0.083387   0.043073   1.936  0.0537 .
## rain          -1.485245   0.920838  -1.613  0.1077
## temp           0.033218   0.014405   2.306  0.0217 *
```

```
## I(FFMC * month_aug) -0.002712 0.001829 -1.483 0.1391
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.388 on 346 degrees of freedom
## Multiple R-squared: 0.04919, Adjusted R-squared: 0.03545
## F-statistic: 3.58 on 5 and 346 DF, p-value: 0.0036
```

```
# Test ANOVA
anova(model3)
```

```
## Analysis of Variance Table
##
## Response: area
##
##           Df Sum Sq Mean Sq F value    Pr(>F)
## month_dec    1  13.98  13.9846    7.2568 0.007408 **
## wind          1   3.82   3.8154    1.9798 0.160305
## rain          1   5.60   5.6017    2.9068 0.089105 .
## temp          1   6.86   6.8580    3.5587 0.060071 .
## I(FFMC * month_aug) 1   4.24   4.2361    2.1982 0.139084
## Residuals    346 666.78   1.9271
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.1 Confronto tra modelli:

```
# Create vector with values
a = c(AIC(model3), BIC(model3), AIC(model2), BIC(model2), AIC(model1), BIC(model1))
```

Akaike Information Criterion (AIC) estimates the in-sample prediction error and indicates the relative

```
# Create vector with nforest
b = c("AIC lm 3", "BIC lm 3", "AIC lm 2", "BIC lm 2", "AIC lm 1", "BIC lm 1")
```

```
# Link the values with nforest
names(a) = b
print(a)
```

```
## AIC lm 3 BIC lm 3 AIC lm 2 BIC lm 2 AIC lm 1 BIC lm 1
## 1237.798 1264.844 1250.011 1350.465 1252.933 1353.461
```

2.2 Previsioni con Model2:

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

```
#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] 1.596855
```

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

##	fit	lwr	upr
## 1	1.0563583	-1.6995064	3.812223
## 5	0.7540603	-1.9949094	3.503030
## 6	1.1626941	-1.5815779	3.906966
## 12	1.1999348	-1.5372090	3.937079
## 18	1.1886159	-1.5492578	3.926490
## 20	0.9094470	-1.8364430	3.655337
## 22	1.0848881	-1.6536671	3.823443
## 27	1.3400658	-1.4020938	4.082225
## 31	1.3692106	-1.3766317	4.115053
## 33	1.0596671	-1.6832339	3.802568
## 35	1.1546462	-1.5853155	3.894608
## 39	1.1501076	-1.5864854	3.886701
## 41	1.2577590	-1.4841748	3.999693
## 43	0.9698926	-1.7779318	3.717717
## 47	1.0137766	-1.7239908	3.751544
## 55	0.8614214	-1.8828183	3.605661
## 57	1.2251558	-1.5128289	3.963141
## 62	0.9333146	-1.8277114	3.694341
## 63	1.0061606	-1.7375691	3.749890
## 67	1.3645353	-1.3776268	4.106697
## 68	1.3266419	-1.4140935	4.067377
## 69	1.4016901	-1.3439371	4.147317
## 71	1.3838643	-1.3703580	4.138087
## 72	1.1134310	-1.6232171	3.850079
## 74	0.9343689	-1.8090689	3.677807
## 76	0.7063382	-2.0486380	3.461314
## 77	1.0053013	-1.7326028	3.743205
## 82	0.7587131	-1.9893600	3.506786
## 85	0.9477885	-1.7941996	3.689777
## 86	1.0145152	-1.7346014	3.763632
## 88	1.1528145	-1.5900827	3.895712
## 89	1.3658887	-1.3797232	4.111501
## 90	1.2118686	-1.5260347	3.949772
## 97	0.7972439	-1.9493688	3.543857
## 98	1.2535620	-1.4860057	3.993130
## 101	1.0854113	-1.6587061	3.829529
## 102	0.9130846	-1.8364519	3.662621
## 103	0.9694144	-1.7731785	3.712007
## 107	1.0883466	-1.6490149	3.825708
## 109	1.1581047	-1.5802173	3.896427
## 113	1.2828434	-1.4579567	4.023644

## 116	1.2654958	-1.4786278	4.009619
## 120	0.8826386	-1.8610212	3.626298
## 124	1.0783812	-1.6589487	3.815711
## 126	1.3598600	-1.3818745	4.101595
## 135	0.9453720	-1.7942472	3.684991
## 136	1.0399911	-1.7077182	3.787700
## 141	1.1028506	-1.6411889	3.846890
## 144	1.2269875	-1.5125071	3.966482
## 146	1.2108219	-1.5359324	3.957576
## 148	1.3293488	-1.4139217	4.072619
## 153	1.1925396	-1.5554412	3.940520
## 159	0.9900239	-1.7516786	3.731726
## 167	0.6019738	-2.1577594	3.361707
## 171	1.1028506	-1.6411889	3.846890
## 173	1.0953768	-1.6486432	3.839397
## 174	0.9966893	-1.7435000	3.736879
## 177	0.9015866	-1.8564232	3.659596
## 180	0.9130846	-1.8364519	3.662621
## 185	1.0701561	-1.6728893	3.813201
## 189	0.7421265	-2.0099780	3.494231
## 191	0.9474770	-1.7936714	3.688625
## 193	1.1605894	-1.5832007	3.904379
## 196	1.2015385	-1.5437283	3.946805
## 200	1.2761998	-1.4642941	4.016694
## 203	0.7010480	-2.0517110	3.453807
## 213	1.3015575	-1.4378052	4.040920
## 214	1.0743208	-1.6713817	3.820023
## 217	0.8950666	-1.8503788	3.640512
## 218	1.1625197	-1.5837305	3.908770
## 224	0.9705931	-1.7684785	3.709665
## 229	1.4238496	-1.3299295	4.177629
## 231	1.4774769	-1.2722585	4.227212
## 234	1.2909772	-1.4527448	4.034699
## 238	1.0332292	-1.7070141	3.773473
## 246	1.3919017	-1.3618185	4.145622
## 247	1.2803135	-1.4667853	4.027412
## 250	0.7269808	-2.0315717	3.485533
## 254	0.9120596	-1.8316094	3.655729
## 260	0.9136843	-1.8340112	3.661380
## 262	0.9826443	-1.7869446	3.752233
## 263	0.9462280	-1.7962137	3.688670
## 264	0.9195194	-1.8228601	3.661899
## 270	0.8351156	-1.9099639	3.580195
## 271	0.9581593	-1.7843500	3.700669
## 272	1.0184302	-1.7236791	3.760539
## 276	2.3383935	-0.6256161	5.302403
## 281	2.2420610	-0.7239890	5.208111
## 282	2.6385865	-0.3105846	5.587758
## 286	1.5279190	-1.2234836	4.279322
## 296	1.0711226	-1.6759890	3.818234
## 297	0.8503929	-1.8939947	3.594780
## 299	1.2849484	-1.4539828	4.023880
## 302	1.3100328	-1.4303127	4.050378
## 304	1.2383064	-1.4993805	3.975993

## 305	1.0092380	-1.7332326	3.751709
## 306	1.0398729	-1.7004184	3.780164
## 308	1.3160616	-1.4264448	4.058568
## 311	1.2908405	-1.4555825	4.037264
## 312	1.1991832	-1.5491739	3.947540
## 313	1.1558760	-1.5906077	3.902360
## 314	1.4410736	-1.3047968	4.186944
## 315	1.0297708	-1.7126315	3.772173
## 318	1.2716612	-1.4669777	4.010300
## 319	1.1262400	-1.6157398	3.868220
## 320	1.4681264	-1.2800163	4.216269
## 325	0.8304620	-1.9146725	3.575597
## 327	1.2843335	-1.4590075	4.027675
## 328	1.1129527	-1.6286385	3.854544
## 330	1.3394509	-1.4027407	4.081642
## 332	1.3147081	-1.4251830	4.054599
## 334	1.0850248	-1.6523233	3.822373
## 337	1.0498383	-1.6905468	3.790223
## 338	1.1727454	-1.5709625	3.916453
## 341	1.0498383	-1.6905468	3.790223
## 347	1.1162745	-1.6254094	3.857959
## 348	0.8609733	-1.8836267	3.605573
## 353	1.2097636	-1.5283112	3.947838
## 354	1.1081408	-1.6307397	3.847021
## 356	1.3248102	-1.4152793	4.064900
## 367	1.2037349	-1.5358496	3.943319
## 369	1.1919377	-1.5459315	3.929807
## 378	0.8820937	-1.8639433	3.628131
## 381	1.4667729	-1.2855399	4.219086
## 384	0.8829769	-1.8609573	3.626911
## 386	0.8545707	-1.8912405	3.600382
## 389	1.0006641	-1.7426709	3.743999
## 399	1.0900144	-1.6549072	3.834936
## 401	1.5306259	-1.2238322	4.285084
## 403	1.1321832	-1.6114372	3.875804
## 408	1.0305223	-1.7074410	3.768486
## 409	1.1448174	-1.5932219	3.882857
## 412	1.2815030	-1.5004097	4.063416
## 415	1.1055412	-1.6371041	3.848187
## 420	1.1342179	-1.6085774	3.877013
## 421	1.2221455	-1.5228643	3.967155
## 423	1.4987612	-1.2583197	4.255842
## 426	0.8379616	-1.9081148	3.584038
## 429	1.3221414	-1.4280818	4.072365
## 430	1.1275742	-1.6150442	3.870193
## 431	1.1699018	-1.5672836	3.907087
## 446	0.9202667	-1.8246983	3.665232
## 448	1.1880140	-1.5766394	3.952667
## 452	0.7804137	-1.9747708	3.535598
## 457	0.7534262	-1.9947223	3.501575
## 458	0.9711779	-1.7706832	3.713039
## 460	1.1588370	-1.6059618	3.923636
## 462	1.0075840	-1.7362662	3.751434
## 463	1.3406807	-1.4137612	4.095123

```
## 469  0.9672713  -1.7719211  3.706464
## 473  1.0338442  -1.7040298  3.771718
## 474  1.2976208  -1.4464937  4.041735
## 478  1.4258180  -1.3220773  4.173713
## 483  1.2288595  -1.5183180  3.976037
## 485  1.4009788  -1.3597001  4.161658
## 486  1.2843075  -1.4688499  4.037465
## 487  1.2932632  -1.4573385  4.043865
## 491  1.2542767  -1.4948155  4.003369
## 493  1.2249177  -1.5354168  3.985252
## 500 -8.2254626 -20.0916986  3.640773
## 503  0.4667698  -2.3585463  3.292086
## 510 -0.7662125  -4.4635771  2.931152
## 513  1.1525889  -1.5940055  3.899183
## 517  0.9924923  -1.7486211  3.733606
```

2.3 LASSO regression:

```
# 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

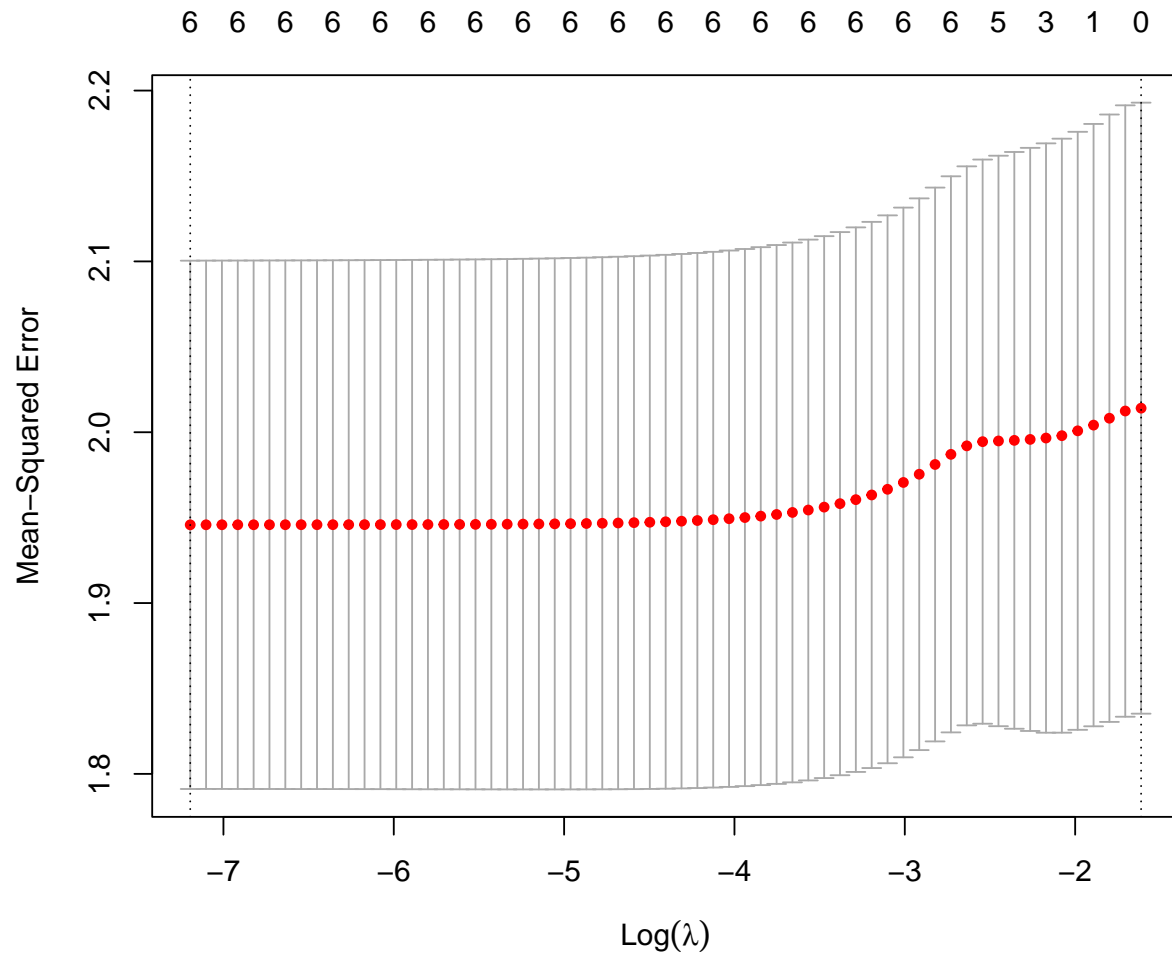
#define matrix of predictor variables (uso solo poche variabili ma potete farlo con tutte da togliere p
x <- data.matrix(train[, c("month_dec","wind","rain","temp","FFMC","month_aug")])

#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
```

```
## [1] 0.0007504308
```

```
#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)
```

```
## 7 x 1 sparse Matrix of class "dgCMatrix"
##              s0
## (Intercept) -0.66874065
## month_dec    1.53364829
## wind         0.08010813
## rain        -1.50538828
## temp         0.02821263
## FPMC         0.01107049
## month_aug    -0.26019301
```

2.4 Previsioni con Lasso:

```
#use lasso regression model to predict response value
new = data.matrix(test[,c("month_dec", "wind", "rain", "temp", "FFMC", "month_aug")])
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))
```

```
## [1] 1.602608
```

Ora potete confrontare modelli diversi con lasso e lm classici, vedete cosa vi dice BIC/AIC e RMSE per decidere quale è il modello ottimale.

2.5 Modello con soglia scelta: logit - glm

```
forest$area2 <- as.factor(ifelse(forest$area>0.5,1,0))

# Regole classiche sono 70% training e 30% test (o 80-20 a vostra scelta)
set.seed(125) # il seme serve per riprodurre le analisi (reproducibilità del codice)
sample <- sample(c(TRUE, FALSE), nrow(forest), replace=TRUE, prob=c(0.7,0.3))
train <- forest[sample, ]
test <- forest[!sample, ]

# Modello con una sola variabile:
model = glm(area2 ~ rain, data = train, family = binomial(link="logit"))
```

```
## Warning: glm.fit: si sono verificate probabilità stimate numericamente pari a 0
## o 1
```

```
# o va bene anche: family="binomial"
summary(model)
```

```
##
## Call:
## glm(formula = area2 ~ rain, family = binomial(link = "logit"),
##      data = train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.04599    0.10724  -0.429   0.668
## rain         -73.92369  3562.53148  -0.021   0.983
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 488.88  on 352  degrees of freedom
## Residual deviance: 482.25  on 351  degrees of freedom
## AIC: 486.25
##
## Number of Fisher Scoring iterations: 16
```



```
# Modello con tutte le variabili:
modell1 = glm(area2 ~ ., data = train, family = binomial(link="logit"))
```

```
## Warning: glm.fit: l'algoritmo non converge
```

```
## Warning: glm.fit: si sono verificate probabilità stimate numericamente pari a 0
## o 1
```

```
summary(modell1)
```

```
##
## Call:
## glm(formula = area2 ~ ., family = binomial(link = "logit"), data = train)
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.179e+02  7.479e+05 -0.001    0.999
## FPMC         2.824e+00  6.219e+03  0.000    1.000
## DMC          -4.230e-01  1.343e+03  0.000    1.000
## DC           3.245e-01  3.106e+02  0.001    0.999
## ISI          7.254e-02  1.145e+04  0.000    1.000
## temp         6.947e+00  4.898e+03  0.001    0.999
## RH           1.534e+00  1.572e+03  0.001    0.999
## wind         6.871e+00  3.230e+03  0.002    0.998
## rain         3.588e+00  1.989e+05  0.000    1.000
## area         1.784e+02  2.636e+04  0.007    0.995
## month_aug    -5.657e+01  4.200e+05  0.000    1.000
## month_dec    -1.850e+02  4.481e+05  0.000    1.000
## month_feb     1.989e+02  4.335e+05  0.000    1.000
## month_jan     2.193e+02  4.645e+05  0.000    1.000
## month_jul    -4.400e+00  4.197e+05  0.000    1.000
## month_jun     3.193e+01  4.392e+05  0.000    1.000
## month_mar     1.769e+02  4.480e+05  0.000    1.000
## month_may    -3.916e+02  5.518e+05 -0.001    0.999
## month_nov          NA          NA          NA          NA
## month_oct    -2.743e+02  6.258e+05  0.000    1.000
## month_sep    -6.737e+01  4.248e+05  0.000    1.000
## day_mon      3.591e+01  4.249e+04  0.001    0.999
## day_sat      6.189e+01  3.421e+04  0.002    0.999
## day_sun      1.663e+01  5.950e+04  0.000    1.000
## day_thu      2.756e+01  4.208e+04  0.001    0.999
## day_tue      4.598e+01  9.720e+04  0.000    1.000
## day_wed      4.357e+01  3.758e+04  0.001    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 4.8888e+02  on 352  degrees of freedom
## Residual deviance: 1.3527e-07  on 327  degrees of freedom
## AIC: 52
##
## Number of Fisher Scoring iterations: 25
```

```

# Removing observations guardando numero dell'osservazione:
train = train[-c(145,142,258),]

# Rimuovere osservazioni basandoci sui valori delle variabili:
# train=train[!(train$temp>300),]

# Modello 1 senza outlier:
model2 = glm(area2 ~ ., data = train, family = binomial(link="logit"))

```

```
## Warning: glm.fit: l'algoritmo non converge
```

```
## Warning: glm.fit: si sono verificate probabilità stimate numericamente pari a 0
## o 1
```

```
summary(model2)
```

```

##
## Call:
## glm(formula = area2 ~ ., family = binomial(link = "logit"), data = train)
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.183e+02  7.527e+05  -0.001    0.999
## FPMC         2.823e+00  6.331e+03   0.000    1.000
## DMC          -4.206e-01  1.364e+03   0.000    1.000
## DC           3.241e-01  3.144e+02   0.001    0.999
## ISI          8.703e-02  1.158e+04   0.000    1.000
## temp         6.944e+00  4.952e+03   0.001    0.999
## RH           1.537e+00  1.596e+03   0.001    0.999
## wind         6.879e+00  3.255e+03   0.002    0.998
## rain         3.379e+00  2.012e+05   0.000    1.000
## area         1.785e+02  2.655e+04   0.007    0.995
## month_aug    -5.658e+01  4.130e+05   0.000    1.000
## month_dec    -1.848e+02  4.421e+05   0.000    1.000
## month_feb     1.991e+02  4.277e+05   0.000    1.000
## month_jan     2.194e+02  4.580e+05   0.000    1.000
## month_jul    -4.279e+00  4.126e+05   0.000    1.000
## month_jun     3.203e+01  4.324e+05   0.000    1.000
## month_mar     1.770e+02  4.413e+05   0.000    1.000
## month_may    -3.918e+02  5.464e+05  -0.001    0.999
## month_nov      NA         NA         NA         NA
## month_oct    -2.705e+02  2.198e+06   0.000    1.000
## month_sep    -6.723e+01  4.179e+05   0.000    1.000
## day_mon       3.586e+01  4.308e+04   0.001    0.999
## day_sat       6.198e+01  3.467e+04   0.002    0.999
## day_sun       1.670e+01  6.042e+04   0.000    1.000
## day_thu       2.764e+01  4.264e+04   0.001    0.999
## day_tue       4.612e+01  9.331e+04   0.000    1.000
## day_wed       4.366e+01  3.815e+04   0.001    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##

```

```
## Null deviance: 4.8447e+02 on 349 degrees of freedom
## Residual deviance: 1.3333e-07 on 324 degrees of freedom
## AIC: 52
##
## Number of Fisher Scoring iterations: 25
```

```
# Modello con variabili scelte da noi (scelte casualmente al momento):
```

```
model3 = glm(area2 ~ month_dec+wind+rain+temp+I(FFMC*month_aug), data = train, family = binomial(link="logit"))
```

```
## Warning: glm.fit: si sono verificate probabilità stimate numericamente pari a 0
## o 1
```

```
summary(model3)
```

```
##
## Call:
## glm(formula = area2 ~ month_dec + wind + rain + temp + I(FFMC *
## month_aug), family = binomial(link = "logit"), data = train)
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.268e+00 5.027e-01 -2.522 0.0117 *
## month_dec 1.774e+01 1.615e+03 0.011 0.9912
## wind 1.037e-01 6.381e-02 1.625 0.1041
## rain -7.350e+01 3.592e+03 -0.020 0.9837
## temp 4.593e-02 2.143e-02 2.144 0.0321 *
## I(FFMC * month_aug) -3.432e-03 2.693e-03 -1.274 0.2026
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 484.47 on 349 degrees of freedom
## Residual deviance: 462.47 on 344 degrees of freedom
## AIC: 474.47
##
## Number of Fisher Scoring iterations: 16
```

```
# per un aumento unitario di variabile_X_modello c'è un aumento del tot% del log dell'odds.
# per un aumento unitario di variabile_X_modello c'è un aumento di exp(valore) = valore_new volte dell'
# (Il vecchio odds va moltiplicato per exp(beta) )
```

```
#use lasso regression model to predict response value
```

```
new = test
```

```
previsioni_mod3 = predict(model3, newdata = new, type="response")
```

```
# la previsione è la probabilità di essere 1!
```

```
prev <- ifelse(previsioni_mod3 > 0.5,"1","0")
```

```
prev<- as.factor(as.vector(prev))
```

```
# Questa tabella mostra previsioni vs valore reale, è chiamata: confusionMatrix
table(prev, test$area2)
```

```
##
## prev  0  1
##      0 53 45
##      1 26 40
```

```
cm <- table(prev, test$area2)

# Indicatori per vedere la bontà del modello:
accuracy <- sum(cm[1], cm[4]) / sum(cm[1:4])
precision <- cm[4] / sum(cm[4], cm[2])
sensitivity <- cm[4] / sum(cm[4], cm[3])
fscore <- (2 * (sensitivity * precision)) / (sensitivity + precision)
specificity <- cm[1] / sum(cm[1], cm[2])

# install.packages("pROC")
library(pROC)
```

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

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Caricamento pacchetto: 'pROC'
```

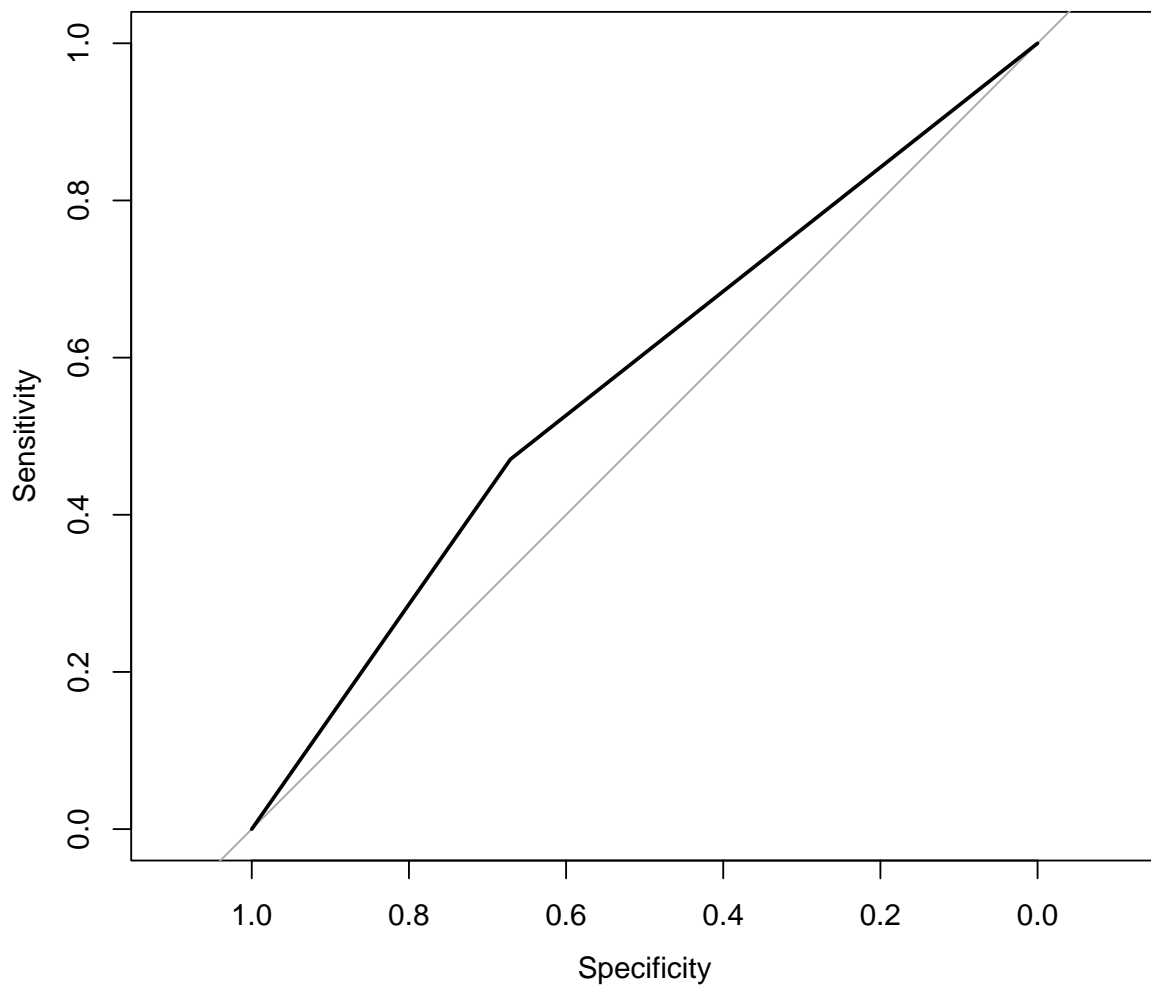
```
## I seguenti oggetti sono mascherati da 'package:stats':
##
##      cov, smooth, var
```

```
roc_object <- roc( as.numeric(test$area2), as.numeric(prev))
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
plot(roc_object)
```



```
# calculate area under curve  
auc(roc_object)
```

```
## Area under the curve: 0.5707
```