Prediction of the Area of Forest Fires Using Data Mining Methods

HU Yin
CHEN Rui
SUN Yidan

2017.12.24

【PART 1 Introduction & Background】

• 1.1 Introduction

The occurrence of forest fires is a significant environmental issue, which affect forest preservation, create economical and ecological damage, and cause human suffering. Such phenomenon is due to multiple causes.

FFMC	The moisture content surface litter and influences ignition and fire spread.	RH	Relative humidity
DMC	The moisture content of shallow organic layers	WIND	Wind speed in km/h
DC	The moisture content of deep organic layers	RAIN	Outside rain in mm/m2
ISI	A score that correlates with fire velocity spread	TEMP	Temperature in Celsius degrees

Figure: Numerical factors: Components of FWI & Temporal Attributes.

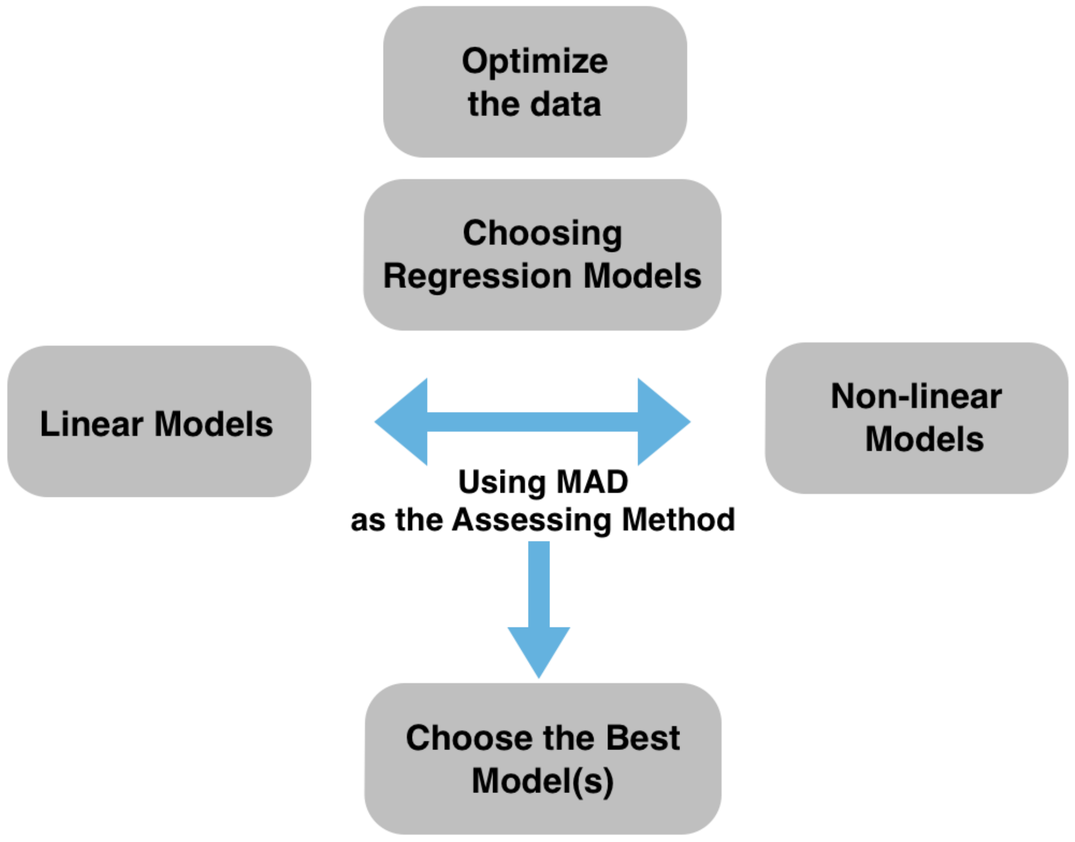
The primary purpose of this study is to to perform a data mining approach (mainly by fitting in regression models) to predict the burned area of forest fires, in the northeast region of Portugal, by using the components from The Canadian Forest Fire Weather Index (FWI) System.

• 1.2 Background

The Canadian Forest Fire Weather Index (FWI) System consists of several important components that account for the effects of fuel moisture and wind on fire behavior: Moreover, training dataset and testing dataset are forehandedly provided.

• 1.3 Methods

This is a flowchart about how we solve the problem:



【PART 2 Before Solving The Problem】

2.1 Preparations & Knowing the data

It is necessary to get familiar with the dataset.

Firstly, In order to obtain a more visual and interpretable training dataset for further analysis, we optimize the data burnt area: By simply observing the data, we find that there are too many 0 values. Therefore, it is preferable to apply the log function y=log(area+1) to the area. The advantage is obvious when doing so: the data become more dispersed, and thus the bias of the model could be reduced.

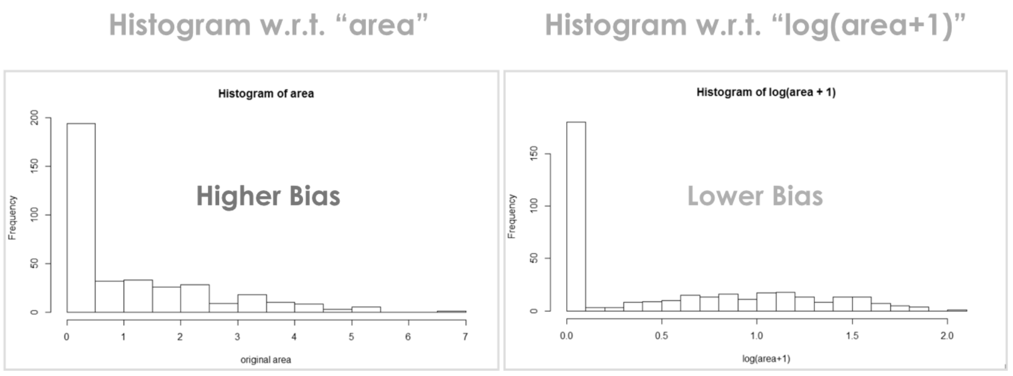
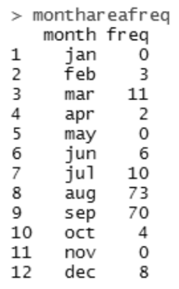
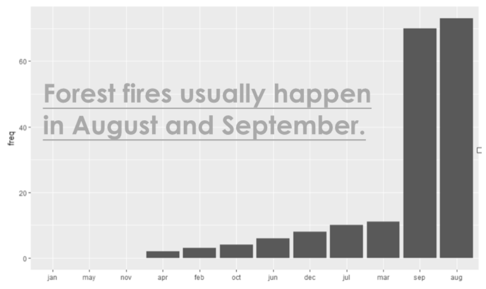


Figure: Histogram w.r.t. area & log(area+1)

Moreover, we want to see how do the non-numerical factors affect the burnt area. The non-numerical factors include month, and the day in the week. We can see that forest fires usually happen in August and September. This is probably because the weather is dryer in these months. And there are more fire cases in the weekend, which is rather obvious to attribute this to the increase of tourists.



```
> monthareafreq
   month freq
1    jan    0
2    feb    3
3    mar   11
4    apr    2
5    may    0
6    jun    6
7    jul   10
8    aug   73
9    sep   70
10   oct    4
11   nov    0
12   dec    8
```



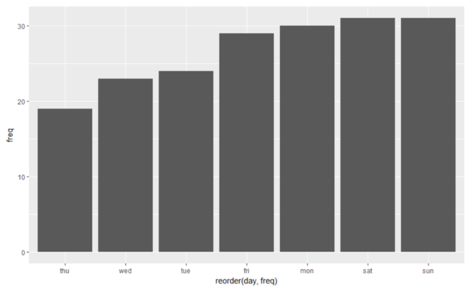


Figure: How do "month" and "the day in a week" affect the burnt area

【PART 3 Data Mining Models】

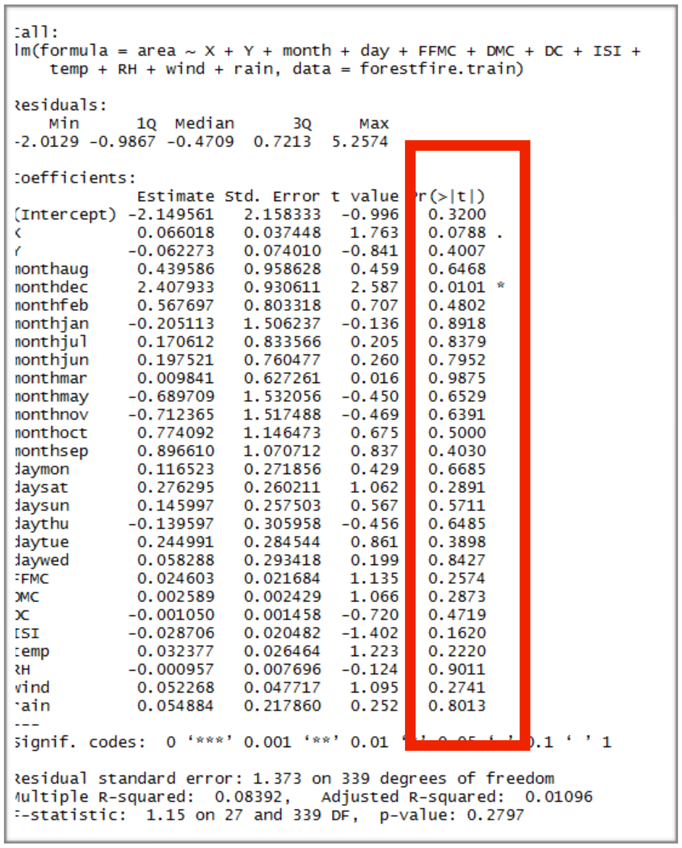
With all preparations done, we can start to fit the models to the data. We know that the regression methods could be divided into linear models and non-linear models. We will look at them separately.

- ### 3.1 Applying Linear Regression Models

  - #### 3.1.1 Methods:

- Method 1: Simple linear Regression

Apply Simple linear Regression model of all the variables to the problem. The P-value of the variables are not significant enough, so it is obvious that this is not a satisfying fit. Therefore, better methods are required for the problem.

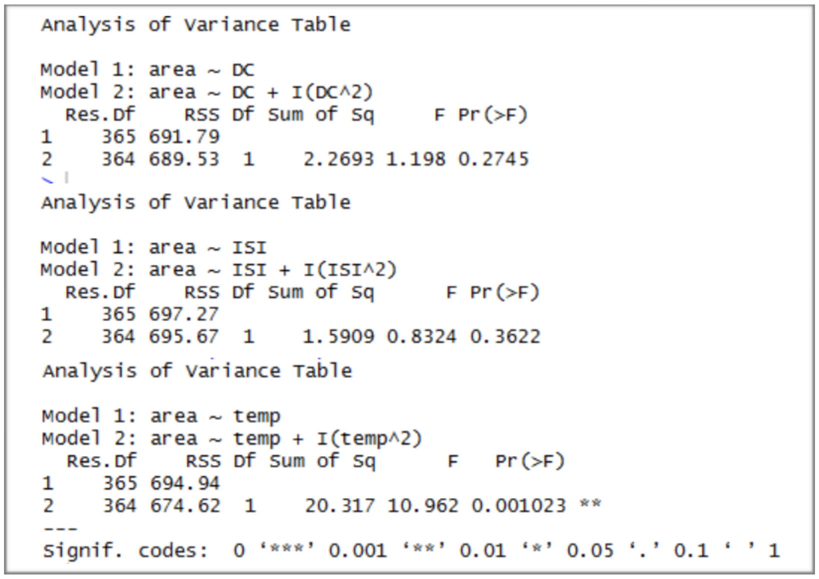


```
:all:
lm(formula = area ~ X + Y + month + day + FFMC + DMC + DC + ISI +
    temp + RH + wind + rain, data = forestfire.train)

Residuals:
    Min      1Q  Median      3Q     Max
-2.0129 -0.9867 -0.4709  0.7213  5.2574

:oefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.149561   2.158333  -0.996   0.3200
<            0.066018   0.037448   1.763   0.0788 .
/           -0.062273   0.074010  -0.841   0.4007
nonthaug     0.439586   0.958628   0.459   0.6468
nonthdec     2.407933   0.930611   2.587   0.0101 *
nonthfeb     0.567697   0.803318   0.707   0.4802
nonthjan    -0.205113   1.506237  -0.136   0.8918
nonthjul     0.170612   0.833566   0.205   0.8379
nonthjun     0.197521   0.760477   0.260   0.7952
nonthmar     0.009841   0.627261   0.016   0.9875
nonthmay    -0.689709   1.532056  -0.450   0.6529
nonthnov    -0.712365   1.517488  -0.469   0.6391
nonthoct     0.774092   1.146473   0.675   0.5000
nonthsep     0.896610   1.070712   0.837   0.4030
daymon       0.116523   0.271856   0.429   0.6685
daysat       0.276295   0.260211   1.062   0.2891
daysun       0.145997   0.257503   0.567   0.5711
daythu      -0.139597   0.305958  -0.456   0.6485
daytue       0.244991   0.284544   0.861   0.3898
daywed       0.058288   0.293418   0.199   0.8427
FFMC         0.024603   0.021684   1.135   0.2574
DMC          0.002589   0.002429   1.066   0.2873
DC          -0.001050   0.001458  -0.720   0.4719
ISI         -0.028706   0.020482  -1.402   0.1620
temp         0.032377   0.026464   1.223   0.2220
RH          -0.000957   0.007696  -0.124   0.9011
wind         0.052268   0.047717   1.095   0.2741
rain         0.054884   0.217860   0.252   0.8013
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.373 on 339 degrees of freedom
Multiple R-squared:  0.08392,   Adjusted R-squared:  0.01096
F-statistic:  1.15 on 27 and 339 DF,  p-value: 0.2797
```

Figure: Summary of simple linear regression

- Method 2: Use anova() for higher-order tests

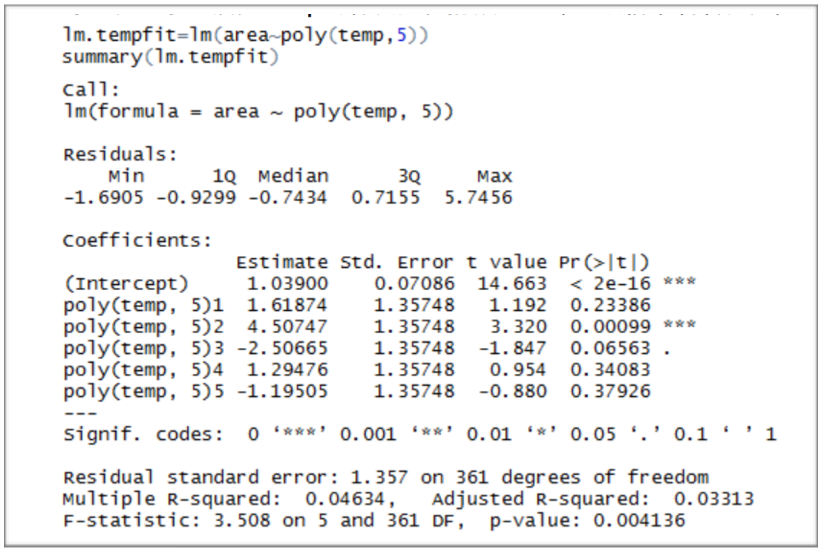
Then, we want to see whether the variables have the tendency to have higher-orders. We use anova to analyze the data. We expect to get a relatively high F-Statistic and a small P-value to illustrate the high possibilities for higher-order terms.



```
Analysis of Variance Table

Model 1: area ~ DC
Model 2: area ~ DC + I(DC^2)
  Res.Df    RSS Df Sum of Sq     F Pr(>F)
1    365 691.79
2    364 689.53  1    2.2693 1.198 0.2745

Analysis of Variance Table

Model 1: area ~ ISI
Model 2: area ~ ISI + I(ISI^2)
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    365 697.27
2    364 695.67  1    1.5909 0.8324 0.3622
Analysis of Variance Table

Model 1: area ~ temp
Model 2: area ~ temp + I(temp^2)
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1    365 694.94
2    364 674.62  1    20.317 10.962 0.001023 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure: extracted anova table

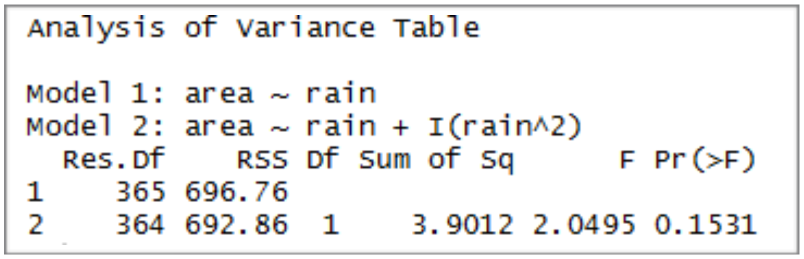
After testing, we find that the factor “temp” has a higher order. And by verifying, its power is two.



```
lm.tempfit=lm(area~poly(temp,5))
summary(lm.tempfit)

Call:
lm(formula = area ~ poly(temp, 5))

Residuals:
    Min      1Q  Median      3Q     Max
-1.6905 -0.9299 -0.7434  0.7155  5.7456

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     1.03900    0.07086  14.663  < 2e-16 ***
poly(temp, 5)1  1.61874    1.35748   1.192  0.23386
poly(temp, 5)2  4.50747    1.35748   3.320  0.00099 ***
poly(temp, 5)3 -2.50665    1.35748  -1.847  0.06563 .
poly(temp, 5)4  1.29476    1.35748   0.954  0.34083
poly(temp, 5)5 -1.19505    1.35748  -0.880  0.37926
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.357 on 361 degrees of freedom
Multiple R-squared:  0.04634,   Adjusted R-squared:  0.03313
F-statistic: 3.508 on 5 and 361 DF,  p-value: 0.004136
```

Figure: polynomial table of the factor “temp”

Besides, “DMC” and “rain” may have higher orders, too. Their tendencies are not obvious so the data is not shown here.



```
Analysis of Variance Table

Model 1: area ~ rain
Model 2: area ~ rain + I(rain^2)
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    365 696.76
2    364 692.86  1    3.9012 2.0495 0.1531
```

Figure: anova table of the factor “rain”

- Method 3: Best subset selection to choose variables

In the next step, we decided to use best subset selection to choose some suitable variables. Notice that there is a precondition: We leave out the interaction terms or higher-order terms, and use simple linear regression to screen for the variables which could relatively fit well.

Since we did not change the variables “month” and “day” into their numeric forms in the previous steps, R will regard 12 months and 7 days as variables. Intuitively, we choose 20 as the upper bound(because 20 is large enough). If the chosen variables contain lots of months and days, it may be difficult to explain the model.

Here, we use the Adjusted R^2, RSS, BIC, and CP methods for selection.

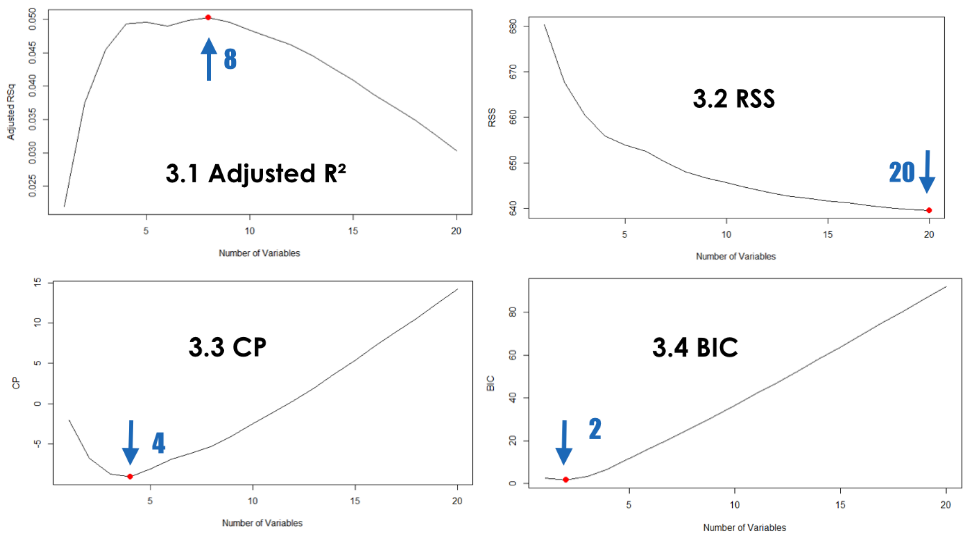
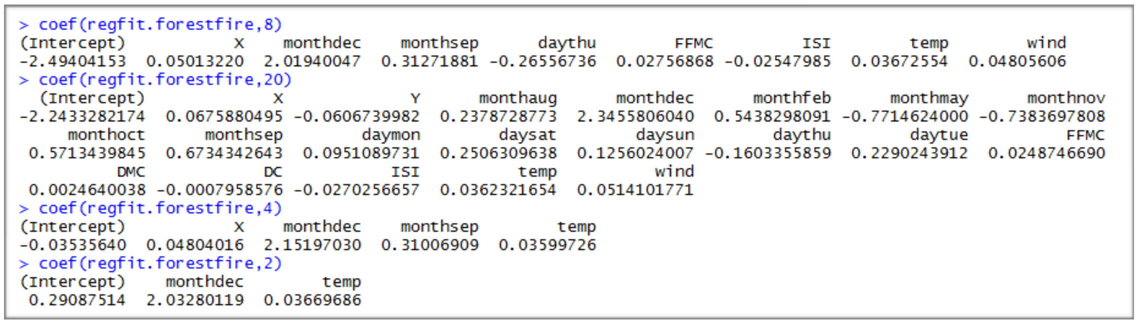


Figure: The result of best subset selection

We analyze the four results and think that 8 is the best number (If the amount of variables is too small, the model can not fit the data well. If the amount of variables is too large, the model can not be well explained).



```
> coef(regfit.forestfire,8)
(Intercept)           X    monthdec    monthsep      daythu        FFMC         ISI        temp        wind 
-2.49404153  0.05013220  2.01940047  0.31271881 -0.26556736  0.02756868 -0.02547985  0.03672554  0.04805606 
> coef(regfit.forestfire,20)
  (Intercept)             X             Y      monthaug      monthdec      monthfeb      monthmay      monthnov 
-2.2433282174  0.0675880495 -0.0606739982  0.2378728773  2.3455806040  0.5438298091 -0.7714624000 -0.7383697808 
     monthoct      monthsep        daymon        daysat        daysun        daythu        daytue          FFMC 
 0.5713439845  0.6734342643  0.0951089731  0.2506309638  0.1256024007 -0.1603355859  0.2290243912  0.0248746690 
          DMC            DC           ISI          temp          wind 
 0.0024640038 -0.0007958576 -0.0270256657  0.0362321654  0.0514101771 
> coef(regfit.forestfire,4)
(Intercept)           X    monthdec    monthsep        temp 
-0.03535640  0.04804016  2.15197030  0.31006909  0.03599726 
> coef(regfit.forestfire,2)
(Intercept)    monthdec        temp 
 0.29087514  2.03280119  0.03669686 
```

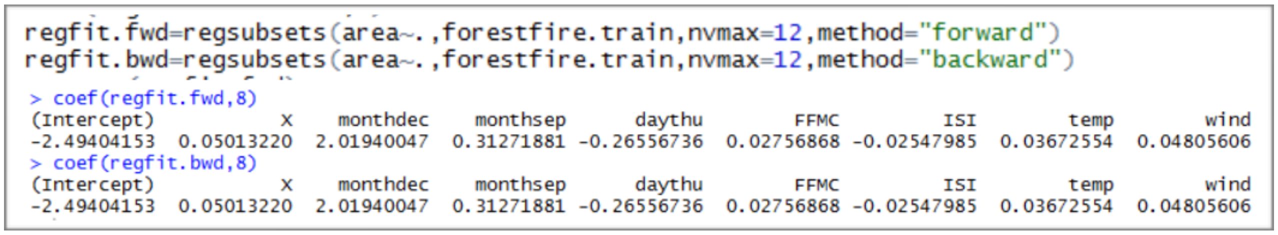
Figure: The coefficient table obtained from the four methods

Therefore, an 8-variable model is preferable. Regression with 8 variables :

X + monthdec + monthsep + daythu + FFMC + ISI + temp + wind

- Method 4: Forward/Backward selection

We further use the forward and backward selection to verify our result.



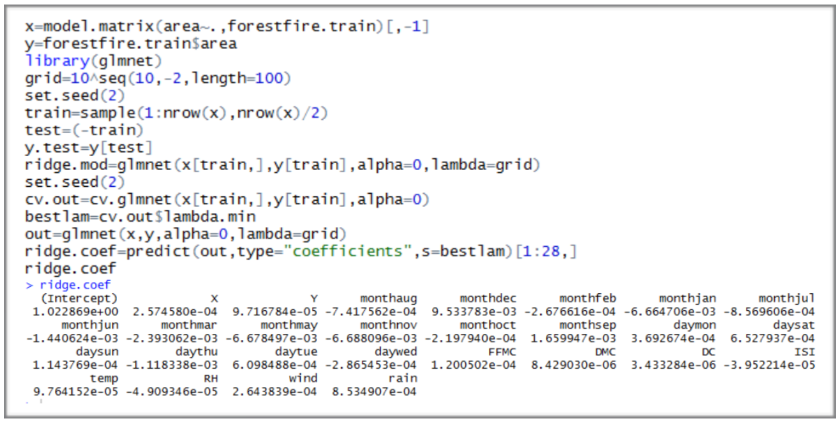
```
regfit.fwd=regsubsets(area~.,forestfire.train,nvmax=12,method="forward")
regfit.bwd=regsubsets(area~.,forestfire.train,nvmax=12,method="backward")
> coef(regfit.fwd,8)
(Intercept)           X    monthdec    monthsep      daythu        FFMC         ISI        temp        wind 
-2.49404153  0.05013220  2.01940047  0.31271881 -0.26556736  0.02756868 -0.02547985  0.03672554  0.04805606 
> coef(regfit.bwd,8)
(Intercept)           X    monthdec    monthsep      daythu        FFMC         ISI        temp        wind 
-2.49404153  0.05013220  2.01940047  0.31271881 -0.26556736  0.02756868 -0.02547985  0.03672554  0.04805606 
```

Figure: The coefficient table obtained from the forward and backward selection

Surprisingly, the results are the same. Moreover, they are also the same with what we have got in the best subset selection. Therefore, basically, we could confirm the variables for the regression model.

- Method 5: Ridge/Lasso

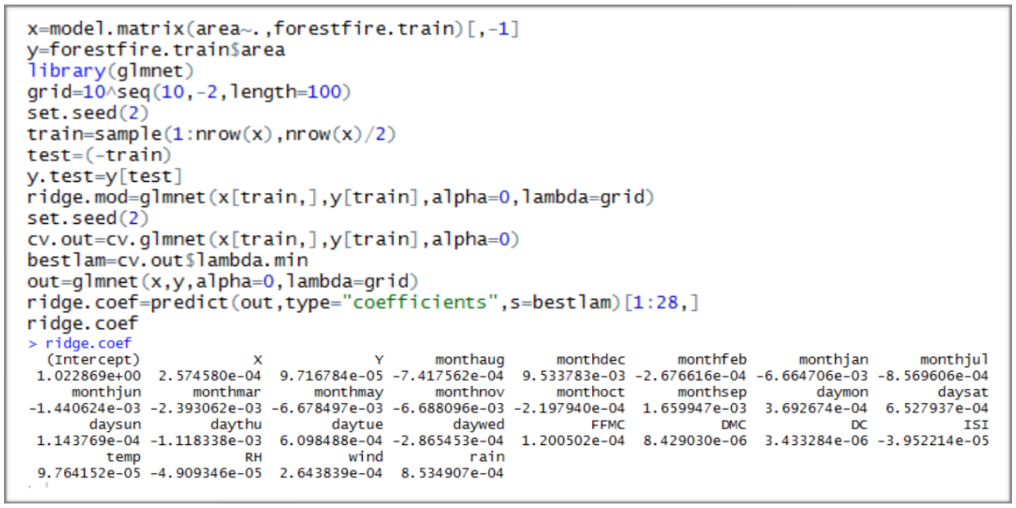
Further, we use Ridge/Lasso regression to seek for more possibilities.



```
x=model.matrix(area~.,forestfire.train)[,-1]
y=forestfire.train$area
library(glmnet)
grid=10^seq(10,-2,length=100)
set.seed(2)
train=sample(1:nrow(x),nrow(x)/2)
test=(-train)
y.test=y[test]
ridge.mod=glmnet(x[train,],y[train],alpha=0,lambda=grid)
set.seed(2)
cv.out=cv.glmnet(x[train,],y[train],alpha=0)
bestlam=cv.out$lambda.min
out=glmnet(x,y,alpha=0,lambda=grid)
ridge.coef=predict(out,type="coefficients",s=bestlam)[1:28,]
ridge.coef
> ridge.coef
  (Intercept)             X             Y     monthaug      monthdec      monthfeb      monthjan      monthjul
 1.022869e+00  2.574580e-04  9.716784e-05 -7.417562e-04  9.533783e-03 -2.676616e-04 -6.664706e-03 -8.569606e-04
     monthjun      monthmar      monthmay      monthnov      monthoct      monthsep        daymon        daysat
-1.440624e-03 -2.393062e-03 -6.678497e-03 -6.688096e-03 -2.197940e-04  1.659947e-03  3.692674e-04  6.527937e-04
       daysun        daythu        daytue        daywed          FFMC           DMC            DC           ISI
 1.143769e-04 -1.118338e-03  6.098488e-04 -2.865453e-04  1.200502e-04  8.429030e-06  3.433284e-06 -3.952214e-05
         temp            RH          wind          rain
 9.764152e-05 -4.909346e-05  2.643839e-04  8.534907e-04
```

Figure: coefficient table of ridge regression

From the coefficient table, we could see that ridge regression gives us really bad fits, and thus it couldn't be adapted.



```
x=model.matrix(area~.,forestfire.train)[,-1]
y=forestfire.train$area
library(glmnet)
grid=10^seq(10,-2,length=100)
set.seed(2)
train=sample(1:nrow(x),nrow(x)/2)
test=(-train)
y.test=y[test]
ridge.mod=glmnet(x[train,],y[train],alpha=0,lambda=grid)
set.seed(2)
cv.out=cv.glmnet(x[train,],y[train],alpha=0)
bestlam=cv.out$lambda.min
out=glmnet(x,y,alpha=0,lambda=grid)
ridge.coef=predict(out,type="coefficients",s=bestlam)[1:28,]
ridge.coef
> ridge.coef
  (Intercept)             X             Y     monthaug      monthdec      monthfeb      monthjan      monthjul
 1.022869e+00  2.574580e-04  9.716784e-05 -7.417562e-04  9.533783e-03 -2.676616e-04 -6.664706e-03 -8.569606e-04
     monthjun      monthmar      monthmay      monthnov      monthoct      monthsep        daymon        daysat
-1.440624e-03 -2.393062e-03 -6.678497e-03 -6.688096e-03 -2.197940e-04  1.659947e-03  3.692674e-04  6.527937e-04
       daysun        daythu        daytue        daywed          FFMC           DMC            DC           ISI
 1.143769e-04 -1.118338e-03  6.098488e-04 -2.865453e-04  1.200502e-04  8.429030e-06  3.433284e-06 -3.952214e-05
         temp            RH          wind          rain
 9.764152e-05 -4.909346e-05  2.643839e-04  8.534907e-04
```

Figure: coefficient table of lasso regression

From the coefficient table, we could see that lasso provides a possible alternative. Screen the non-zero factors out, we found that the following six variables are relatively significant in the model:

X + monthdec + monthsep + DC + temp + RH

- **3.1.2 Comparing the models**

Now that we have two candidates, which were derived by Adjusted R^2/Forward/Backward selection and Lasso, respectively.

1. X + monthdec + monthsep + daythu + FFMC + ISI + temp + wind
2. X + monthdec + monthsep + DC + temp + RH

【The measurement of models】

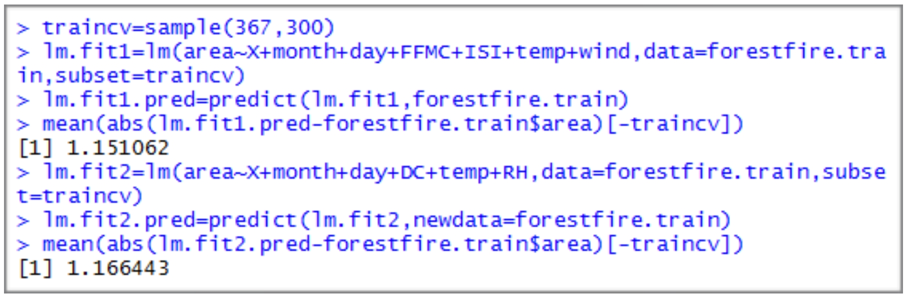
Here, we use MEAN ABSOLUTE DEVIATION (or MAD/RMSE) as the measurement of a model.

$$MAD \quad = 1/N \times \sum_{i=1}^{N} |y_i - \widehat{y_i}|$$
$$RMSE = \sqrt{\sum_{i=1}^{N} (y_i - \widehat{y_i})^2/N}$$

We could see that the result are quite similar. Therefore we consider that both of them are adaptable under the same conditions.

【Method of choosing models】

We divide the train.csv into two parts. The first part is the training part while the second one is the test part. We use training part to train the model and use test part to select the model. The result is almost the model. But the first model is a little bit better (The test MAD uses log(area+1) as the responser).



```
> traincv=sample(367,300)
> lm.fit1=lm(area~X+month+day+FFMC+ISI+temp+wind,data=forestfire.tra
in,subset=traincv)
> lm.fit1.pred=predict(lm.fit1,forestfire.train)
> mean(abs(lm.fit1.pred-forestfire.train$area)[-traincv])
[1] 1.151062
> lm.fit2=lm(area~X+month+day+DC+temp+RH,data=forestfire.train,subse
t=traincv)
> lm.fit2.pred=predict(lm.fit2,newdata=forestfire.train)
> mean(abs(lm.fit2.pred-forestfire.train$area)[-traincv])
[1] 1.166443
```

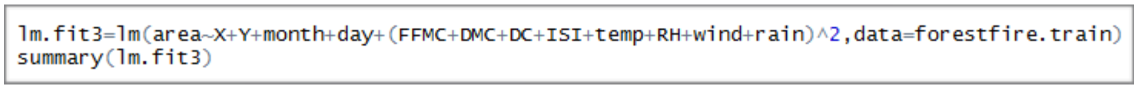
Figure: Using validation methods to assess the two models

- **3.1.3 Considering Interaction terms**

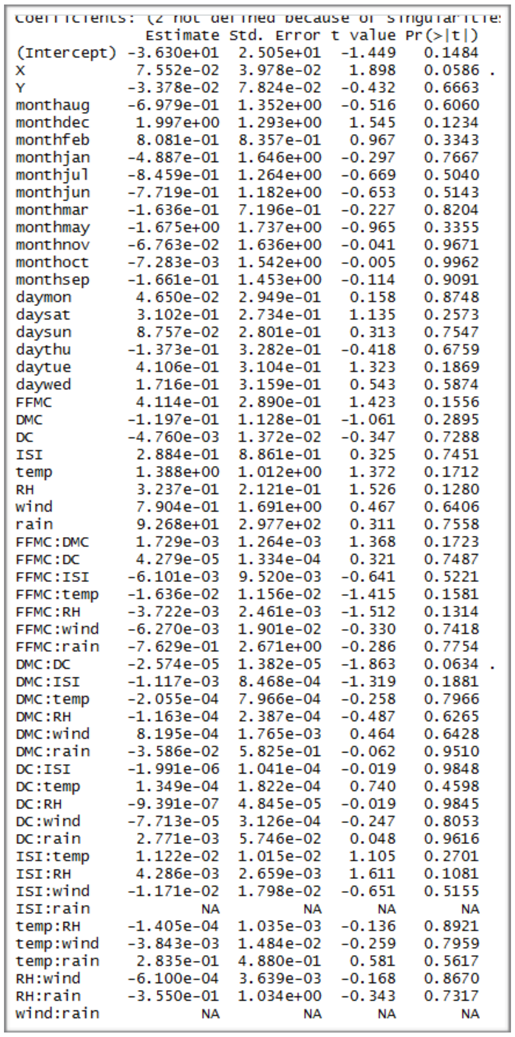
Now that we want to see whether there could be interaction terms in this problem.

Step1:

We print out the relationships between ALL numerical variables. In this case, we only find the term DMC:DC could possibly be a useful interaction term in this model.



```
lm.fit3=lm(area~X+Y+month+day+(FFMC+DMC+DC+ISI+temp+RH+wind+rain)^2,data=forestfire.train)
summary(lm.fit3)
```

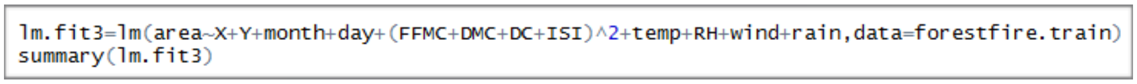


```
Coefficients: (2 not defined because of singularitie
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.630e+01  2.505e+01  -1.449   0.1484
X            7.552e-02  3.978e-02   1.898   0.0586 .
Y           -3.378e-02  7.824e-02  -0.432   0.6663
monthaug    -6.979e-01  1.352e+00  -0.516   0.6060
monthdec     1.997e+00  1.293e+00   1.545   0.1234
monthfeb     8.081e-01  8.357e-01   0.967   0.3343
monthjan    -4.887e-01  1.646e+00  -0.297   0.7667
monthjul    -8.459e-01  1.264e+00  -0.669   0.5040
monthjun    -7.719e-01  1.182e+00  -0.653   0.5143
monthmar    -1.636e-01  7.196e-01  -0.227   0.8204
monthmay    -1.675e+00  1.737e+00  -0.965   0.3355
monthnov    -6.763e-02  1.636e+00  -0.041   0.9671
monthoct    -7.283e-03  1.542e+00  -0.005   0.9962
monthsep    -1.661e-01  1.453e+00  -0.114   0.9091
daymon       4.650e-02  2.949e-01   0.158   0.8748
daysat       3.102e-01  2.734e-01   1.135   0.2573
daysun       8.757e-02  2.801e-01   0.313   0.7547
daythu      -1.373e-01  3.282e-01  -0.418   0.6759
daytue       4.106e-01  3.104e-01   1.323   0.1869
daywed       1.716e-01  3.159e-01   0.543   0.5874
FFMC         4.114e-01  2.890e-01   1.423   0.1556
DMC         -1.197e-01  1.128e-01  -1.061   0.2895
DC          -4.760e-03  1.372e-02  -0.347   0.7288
ISI          2.884e-01  8.861e-01   0.325   0.7451
temp         1.388e+00  1.012e+00   1.372   0.1712
RH           3.237e-01  2.121e-01   1.526   0.1280
wind         7.904e-01  1.691e+00   0.467   0.6406
rain         9.268e+01  2.977e+02   0.311   0.7558
FFMC:DMC     1.729e-03  1.264e-03   1.368   0.1723
FFMC:DC      4.279e-05  1.334e-04   0.321   0.7487
FFMC:ISI    -6.101e-03  9.520e-03  -0.641   0.5221
FFMC:temp   -1.636e-02  1.156e-02  -1.415   0.1581
FFMC:RH     -3.722e-03  2.461e-03  -1.512   0.1314
FFMC:wind   -6.270e-03  1.901e-02  -0.330   0.7418
FFMC:rain   -7.629e-01  2.671e+00  -0.286   0.7754
DMC:DC      -2.574e-05  1.382e-05  -1.863   0.0634 .
DMC:ISI     -1.117e-03  8.468e-04  -1.319   0.1881
DMC:temp    -2.055e-04  7.966e-04  -0.258   0.7966
DMC:RH      -1.163e-04  2.387e-04  -0.487   0.6265
DMC:wind     8.195e-04  1.765e-03   0.464   0.6428
DMC:rain    -3.586e-02  5.825e-01  -0.062   0.9510
DC:ISI      -1.991e-06  1.041e-04  -0.019   0.9848
DC:temp      1.349e-04  1.822e-04   0.740   0.4598
DC:RH       -9.391e-07  4.845e-05  -0.019   0.9845
DC:wind     -7.713e-05  3.126e-04  -0.247   0.8053
DC:rain      2.771e-03  5.746e-02   0.048   0.9616
ISI:temp     1.122e-02  1.015e-02   1.105   0.2701
ISI:RH       4.286e-03  2.659e-03   1.611   0.1081
ISI:wind    -1.171e-02  1.798e-02  -0.651   0.5155
ISI:rain            NA         NA      NA       NA
temp:RH     -1.405e-04  1.035e-03  -0.136   0.8921
temp:wind   -3.843e-03  1.484e-02  -0.259   0.7959
temp:rain    2.835e-01  4.880e-01   0.581   0.5617
RH:wind     -6.100e-04  3.639e-03  -0.168   0.8670
RH:rain     -3.550e-01  1.034e+00  -0.343   0.7317
wind:rain           NA         NA      NA       NA
```

Figure: Summary table of the pairwise interaction terms

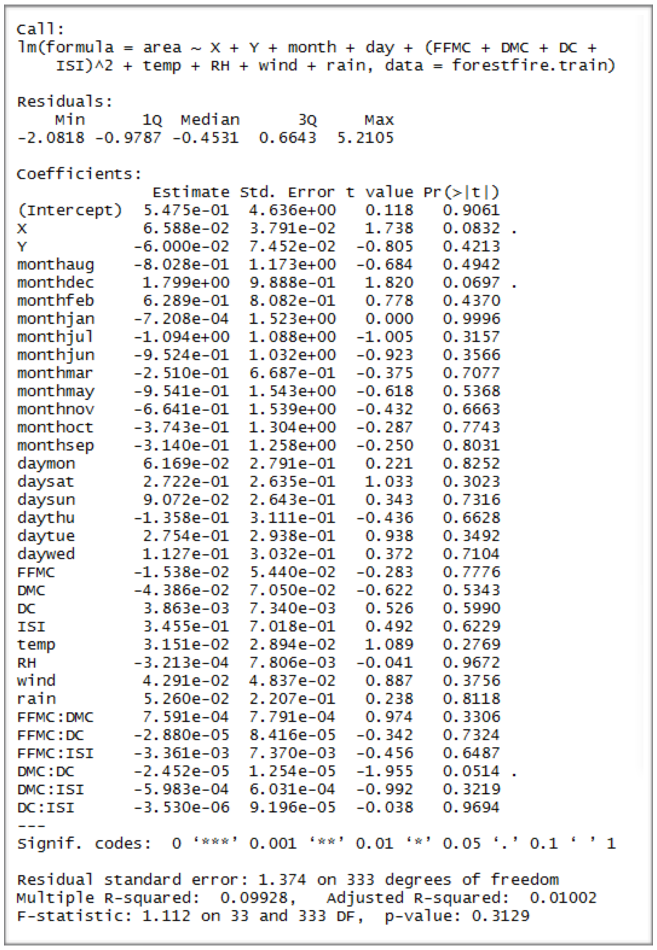
Step 2:

Analyze the relationships between components of FWI (FFMC, DMC, DC, ISI)



```
lm.fit3=lm(area~X+Y+month+day+(FFMC+DMC+DC+ISI)^2+temp+RH+wind+rain,data=forestfire.train)
summary(lm.fit3)
```

It is previously mentioned that the numerical variables are consist of the Components of FWI, and Temporal Attributes. For step two, we analyze the relationships between components of FWI. The new model also told us that only DMC:DC could possibly be a useful interaction term.



```
Call:
lm(formula = area ~ X + Y + month + day + (FFMC + DMC + DC +
    ISI)^2 + temp + RH + wind + rain, data = forestfire.train)

Residuals:
    Min      1Q  Median      3Q     Max
-2.0818 -0.9787 -0.4531  0.6643  5.2105

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.475e-01  4.636e+00   0.118   0.9061
X            6.588e-02  3.791e-02   1.738   0.0832 .
Y           -6.000e-02  7.452e-02  -0.805   0.4213
monthaug    -8.028e-01  1.173e+00  -0.684   0.4942
monthdec     1.799e+00  9.888e-01   1.820   0.0697 .
monthfeb     6.289e-01  8.082e-01   0.778   0.4370
monthjan    -7.208e-04  1.523e+00   0.000   0.9996
monthjul    -1.094e+00  1.088e+00  -1.005   0.3157
monthjun    -9.524e-01  1.032e+00  -0.923   0.3566
monthmar    -2.510e-01  6.687e-01  -0.375   0.7077
monthmay    -9.541e-01  1.543e+00  -0.618   0.5368
monthnov    -6.641e-01  1.539e+00  -0.432   0.6663
monthoct    -3.743e-01  1.304e+00  -0.287   0.7743
monthsep    -3.140e-01  1.258e+00  -0.250   0.8031
daymon       6.169e-02  2.791e-01   0.221   0.8252
daysat       2.722e-01  2.635e-01   1.033   0.3023
daysun       9.072e-02  2.643e-01   0.343   0.7316
daythu      -1.358e-01  3.111e-01  -0.436   0.6628
daytue       2.754e-01  2.938e-01   0.938   0.3492
daywed       1.127e-01  3.032e-01   0.372   0.7104
FFMC        -1.538e-02  5.440e-02  -0.283   0.7776
DMC         -4.386e-02  7.050e-02  -0.622   0.5343
DC           3.863e-03  7.340e-03   0.526   0.5990
ISI          3.455e-01  7.018e-01   0.492   0.6229
temp         3.151e-02  2.894e-02   1.089   0.2769
RH          -3.213e-04  7.806e-03  -0.041   0.9672
wind         4.291e-02  4.837e-02   0.887   0.3756
rain         5.260e-02  2.207e-01   0.238   0.8118
FFMC:DMC     7.591e-04  7.791e-04   0.974   0.3306
FFMC:DC     -2.880e-05  8.416e-05  -0.342   0.7324
FFMC:ISI    -3.361e-03  7.370e-03  -0.456   0.6487
DMC:DC      -2.452e-05  1.254e-05  -1.955   0.0514 .
DMC:ISI     -5.983e-04  6.031e-04  -0.992   0.3219
DC:ISI      -3.530e-06  9.196e-05  -0.038   0.9694
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.374 on 333 degrees of freedom
Multiple R-squared:  0.09928,	Adjusted R-squared:  0.01002
F-statistic: 1.112 on 33 and 333 DF,  p-value: 0.3129
```

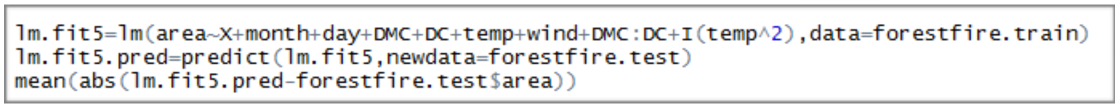
Step 3: Modify the predictors

According to all what we have already obtained, we:

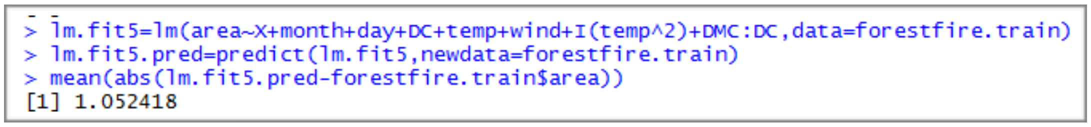
1. raise the temperature term to its quadratic form

2. add in the interaction term DMC:DC

3. delete some variables with bad fits.

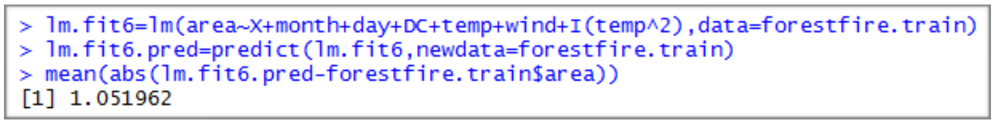


```
lm.fit5=lm(area~X+month+day+DMC+DC+temp+wind+DMC:DC+I(temp^2),data=forestfire.train)
lm.fit5.pred=predict(lm.fit5,newdata=forestfire.test)
mean(abs(lm.fit5.pred-forestfire.test$area))
```



```
> lm.fit5=lm(area~X+month+day+DC+temp+wind+I(temp^2)+DMC:DC,data=forestfire.train)
> lm.fit5.pred=predict(lm.fit5,newdata=forestfire.train)
> mean(abs(lm.fit5.pred-forestfire.train$area))
[1] 1.052418
```

The MAD result is good. However if we remove the interaction term, we even obtained a better result. This is the best result up to now!

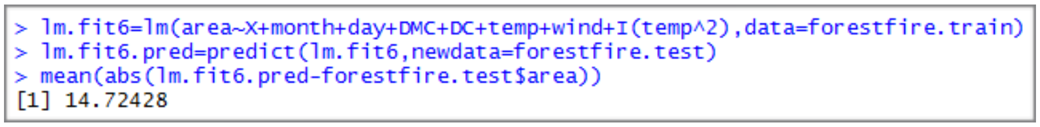


```
> lm.fit6=lm(area~X+month+day+DC+temp+wind+I(temp^2),data=forestfire.train)
> lm.fit6.pred=predict(lm.fit6,newdata=forestfire.train)
> mean(abs(lm.fit6.pred-forestfire.train$area))
[1] 1.051962
```

We discussed some other situations. Including exchanging RH and wind, changing the interaction term of DC and DMC and deleting DC or DMC. Their results are not satisfying. Therefore we choose the regression model of variables:

X + month + day + DC + temp + wind+(temp)^2

and use test.csv to calculate the MAD(Now we use area as the responser).



```
> lm.fit6=lm(area~X+month+day+DMC+DC+temp+wind+I(temp^2),data=forestfire.train)
> lm.fit6.pred=predict(lm.fit6,newdata=forestfire.test)
> mean(abs(lm.fit6.pred-forestfire.test$area))
[1] 14.72428
```

Periodical conclusion for part 3.1: Model

X + month + day + DC + temp + wind+(temp)^2

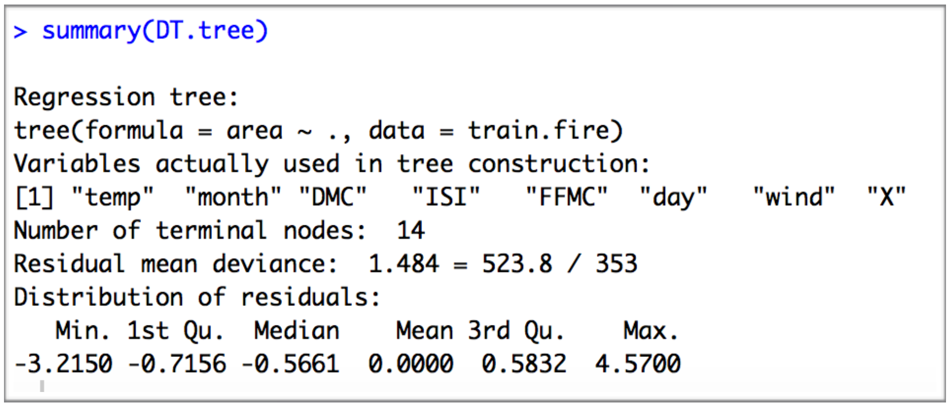
is considered to be the best.

- **3.2 Applying Non-Linear Regression Models**

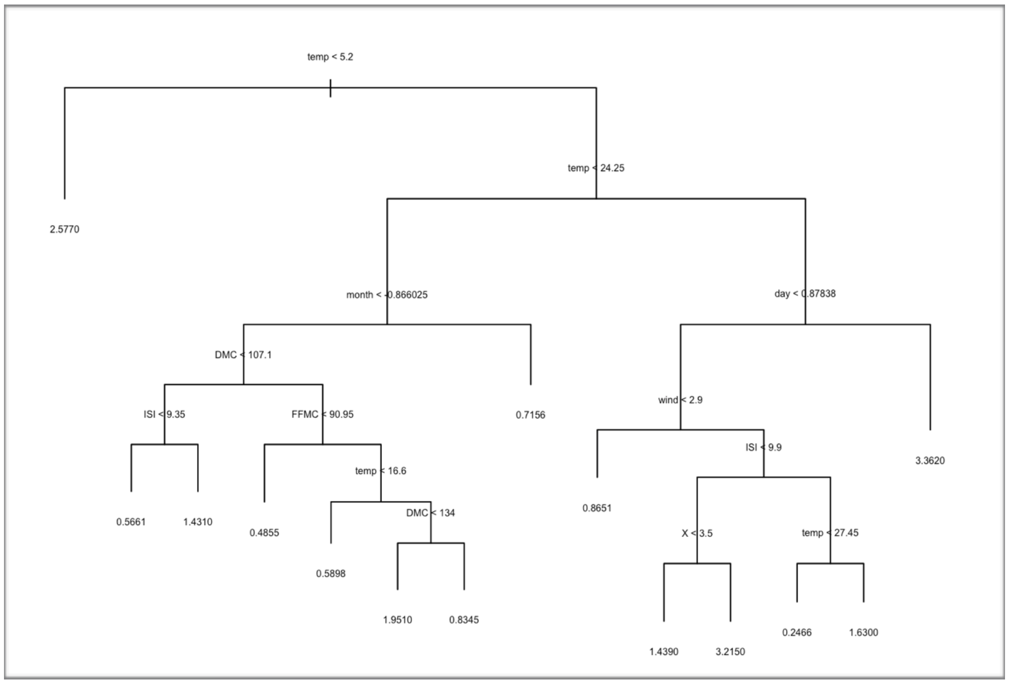
- **3.2.1 Methods:**

- Method 1: Regression Tree:

Now, we fit a regression tree with all the attributes.



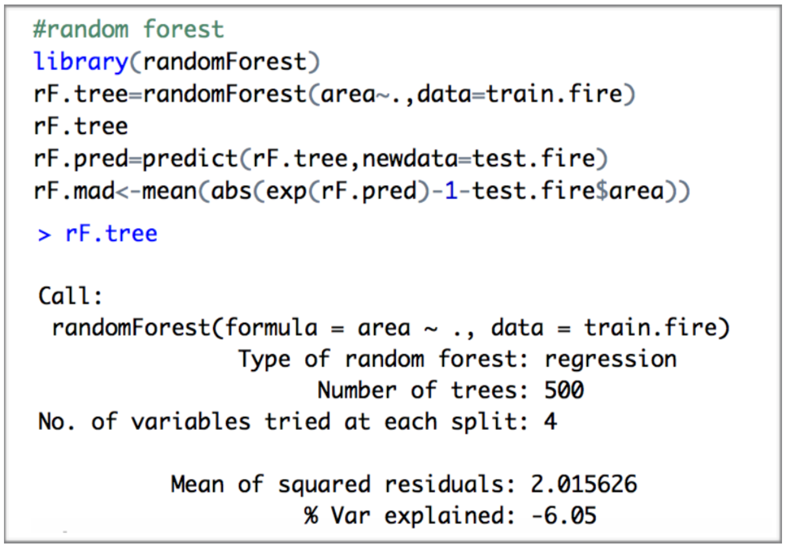
```
> summary(DT.tree)

Regression tree:
tree(formula = area ~ ., data = train.fire)
Variables actually used in tree construction:
[1] "temp"  "month" "DMC"   "ISI"   "FFMC"  "day"   "wind"  "X"
Number of terminal nodes:  14
Residual mean deviance:  1.484 = 523.8 / 353
Distribution of residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-3.2150 -0.7156 -0.5661  0.0000  0.5832  4.5700
```



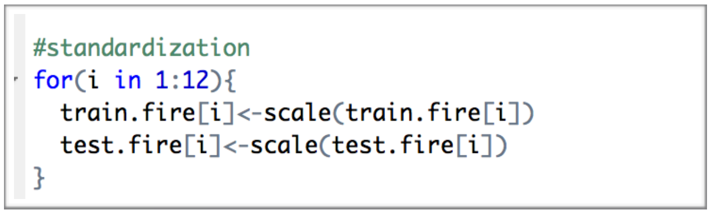
Figures: Summary of tree method & tree diagram

• Method 2: Random Forest:

Apply random forest from the package randomForest with default parameters(T=500). It is worth noticing that it is necessary to standardize all the attributes for further comparisons.



```
#random forest
library(randomForest)
rF.tree=randomForest(area~.,data=train.fire)
rF.tree
rF.pred=predict(rF.tree,newdata=test.fire)
rF.mad<-mean(abs(exp(rF.pred)-1-test.fire$area))
> rF.tree

Call:
 randomForest(formula = area ~ ., data = train.fire)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 2.015626
                    % Var explained: -6.05
```



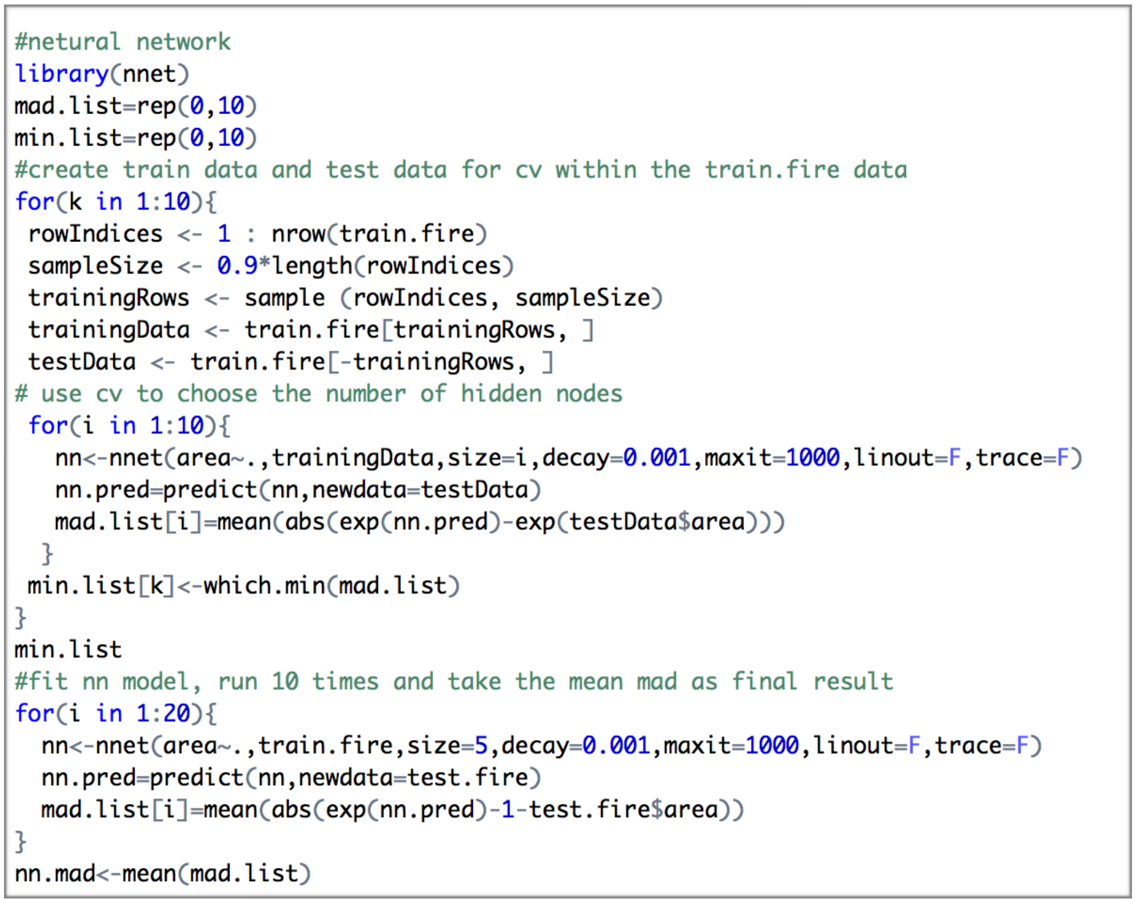
```
#standardization
for(i in 1:12){
  train.fire[i]<-scale(train.fire[i])
  test.fire[i]<-scale(test.fire[i])
}
```

Figures: Random forest method & standardization.

• Method 3: Neural network

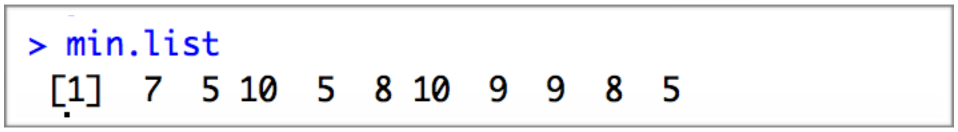
Fit feed-forward neural network model with package nnet, which have multilayer perceptrons with one hidden layer of H hidden nodes and logistic activation functions and one output node.

Use 90% random sample from the train fire as train data and the left part as test data. Run 10 times with the different sample, for each time fit 10 models with number of nodes from 1 to 10 and record the number which lead to lowest MAD. For the 10 numbers obtained, choose the one which appear most fluently as the final choice (H=5)



```
#netural network
library(nnet)
mad.list=rep(0,10)
min.list=rep(0,10)
#create train data and test data for cv within the train.fire data
for(k in 1:10){
 rowIndices <- 1 : nrow(train.fire)
 sampleSize <- 0.9*length(rowIndices)
 trainingRows <- sample (rowIndices, sampleSize)
 trainingData <- train.fire[trainingRows, ]
 testData <- train.fire[-trainingRows, ]
# use cv to choose the number of hidden nodes
 for(i in 1:10){
   nn<-nnet(area~.,trainingData,size=i,decay=0.001,maxit=1000,linout=F,trace=F)
   nn.pred=predict(nn,newdata=testData)
   mad.list[i]=mean(abs(exp(nn.pred)-exp(testData$area)))
  }
 min.list[k]<-which.min(mad.list)
}
min.list
#fit nn model, run 10 times and take the mean mad as final result
for(i in 1:20){
  nn<-nnet(area~.,train.fire,size=5,decay=0.001,maxit=1000,linout=F,trace=F)
  nn.pred=predict(nn,newdata=test.fire)
  mad.list[i]=mean(abs(exp(nn.pred)-1-test.fire$area))
}
nn.mad<-mean(mad.list)
```

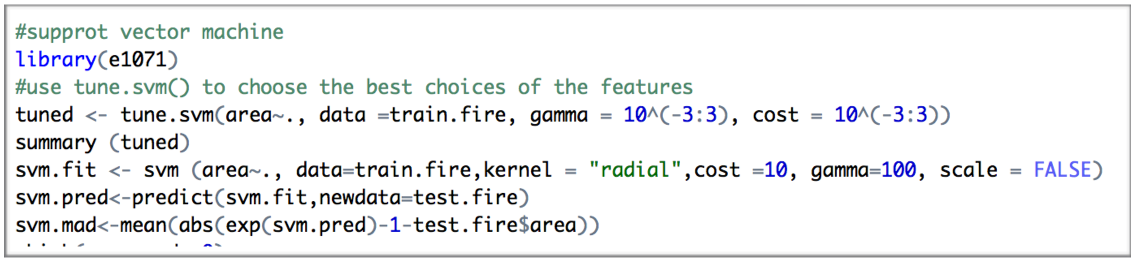
Figures: Neural network method



```
> min.list
 [1]  7  5 10  5  8 10  9  9  8  5
```

• Method 4: Support Vector Machine

Fit support vector machine(SVM) with package e1071, use tune.svm() to choose the best choices of the features(gamma=100,cost=10), For the kernel function, choose the radial basis function.

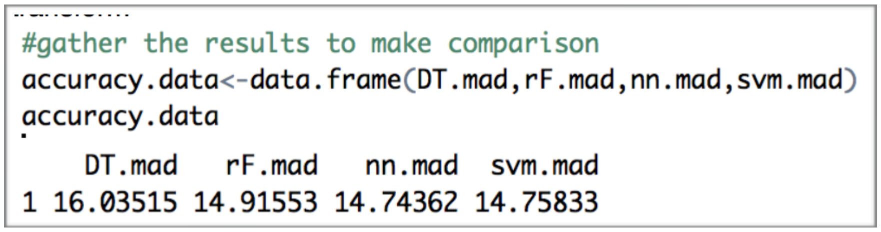


```
#supprot vector machine
library(e1071)
#use tune.svm() to choose the best choices of the features
tuned <- tune.svm(area~., data =train.fire, gamma = 10^(-3:3), cost = 10^(-3:3))
summary (tuned)
svm.fit <- svm (area~., data=train.fire,kernel = "radial",cost =10, gamma=100, scale = FALSE)
svm.pred<-predict(svm.fit,newdata=test.fire)
svm.mad<-mean(abs(exp(svm.pred)-1-test.fire$area))
```

Figure: Support vector machine

- **3.2.2 Comparing the models**

  Collect MADs which are processed with the inverse of the logarithm transform:



```
#gather the results to make comparison
accuracy.data<-data.frame(DT.mad,rF.mad,nn.mad,svm.mad)
accuracy.data

    DT.mad   rF.mad   nn.mad  svm.mad
1 16.03515 14.91553 14.74362 14.75833
```

Figure: MAD comparisons

Periodical conclusion for part 3.2:

Generally, svm and neural network model lead to the lowest MAD followed by random forest while the decision tree has the worst performance. It is worth mentioning that the neural network will lead to prediction for the area with value 0, while other models fail to do so.

【PART 4 Conclusions】

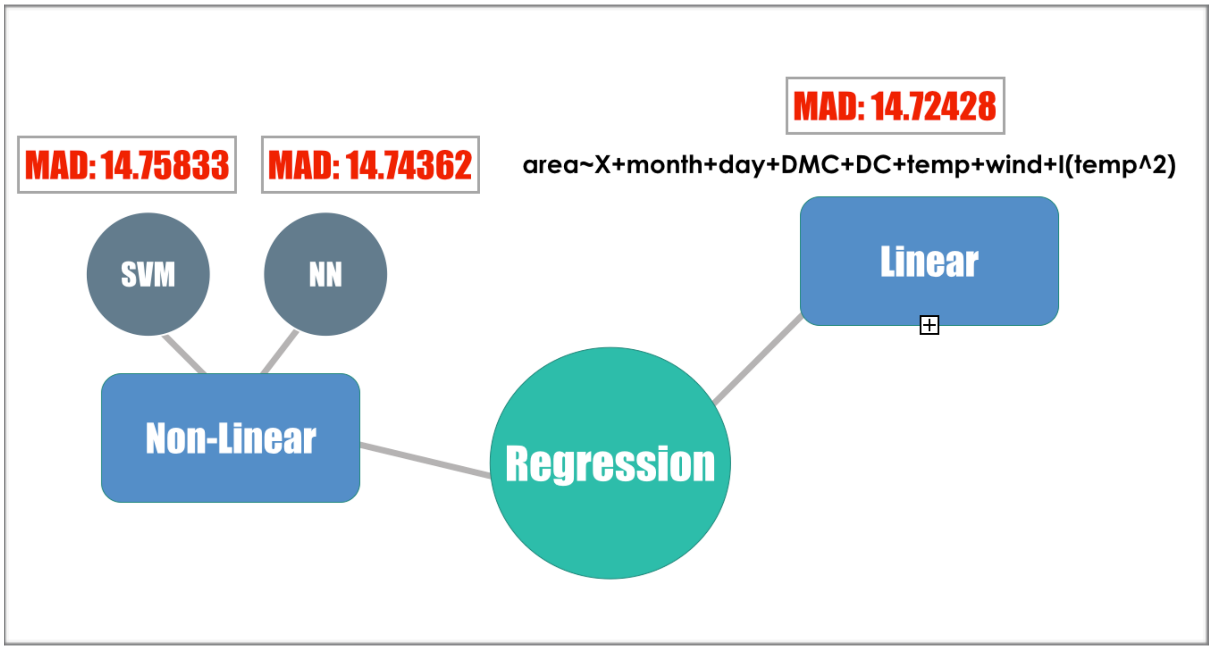


Figure: Map of the best results of linear regression models & non-linear models

To conclude with, we have fit linear regression models and non-linear models to the problem, from both of which we obtained good results. The regression model :

X + month + day + DC + temp + wind+(temp)^2

gives us a MAD of 14.72, which is the best result among what we have all found.