Assignment 5

Ilaria Crippa

2024-02-18

Contents

1 Assignment 5 **1**

1.1 Variable selection and cross validation 1

1.1.1 Best subset selection 1

1.1.2 Best model according to AIC, BIC, adjusted R^2 and Mallow's $\mathrm{C_p}$ 3

1.1.3 Backward selection 5

1.1.4 Forward selection 6

1.1.5 Validation set approach 7

1.1.6 LOOCV cross-validation approach 8

1 Assignment 5

1.1 Variable selection and cross validation

1.1.1 Best subset selection

We consider all the covariates present in the dataset to explain the response and assume a linear regression model. To explore all the possible models, we perform a best subset selection, which is a technique to understand which covariates are needed to fit the best model among all the possible models fitted with a pre-defined number of predictors.

```r
ols = regsubsets (
  Next_Tmax  ~ RHmin + RHmax + Tmax_lapse + Tmin_lapse + WS + LH + lat + lon +
  Slope + Solar.radiation + Next_Tmin + grouped_station + CC + PPT,
  data = data,
  nvmax = 16
)
summary = summary(ols)
summary
```

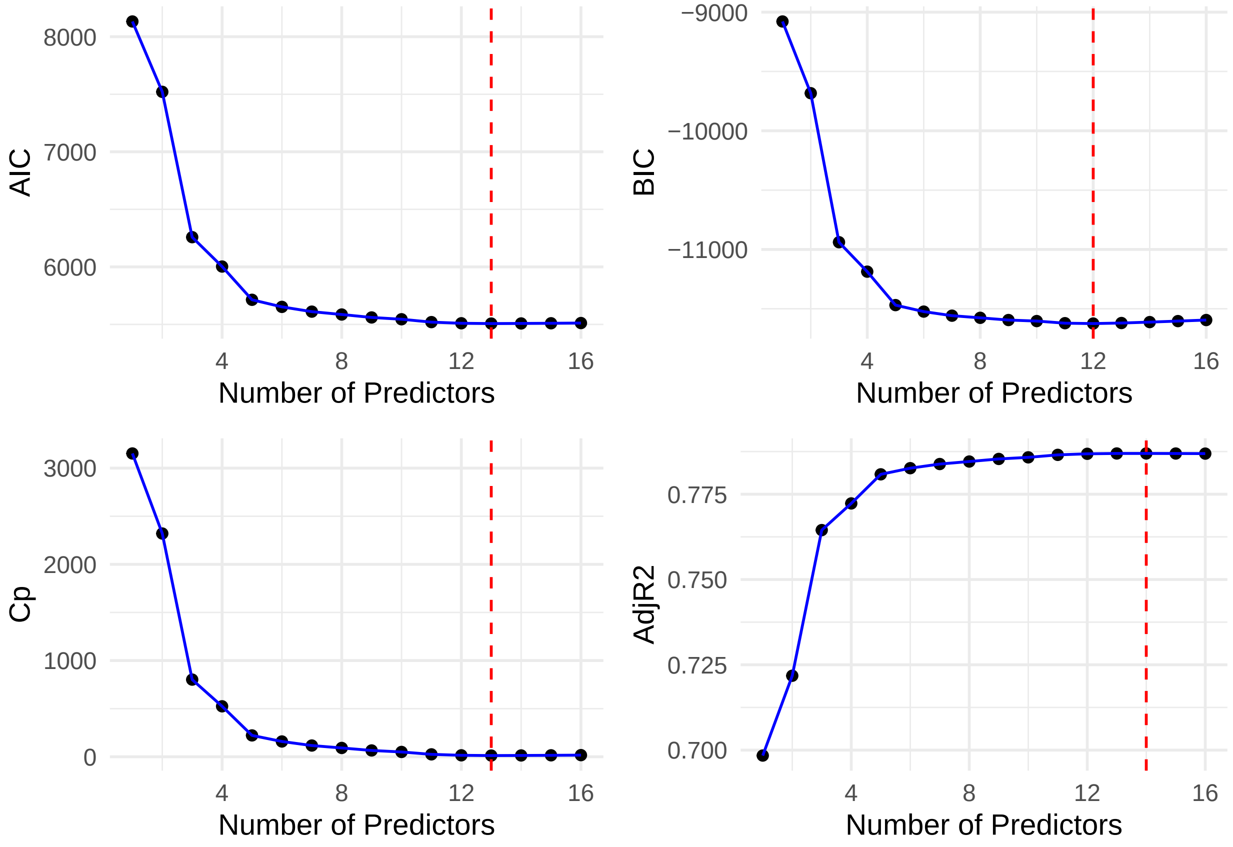
```
## Subset selection object
## Call: regsubsets.formula(Next_Tmax ~ RHmin + RHmax + Tmax_lapse + Tmin_lapse +
##     WS + LH + lat + lon + Slope + Solar.radiation + Next_Tmin +
##     grouped_station + CC + PPT, data = data, nvmax = 16)
## 16 Variables  (and intercept)
##                                    Forced in Forced out
## RHmin                                  FALSE      FALSE
## RHmax                                  FALSE      FALSE
## Tmax_lapse                             FALSE      FALSE
## Tmin_lapse                             FALSE      FALSE
## WS                                     FALSE      FALSE
## LH                                     FALSE      FALSE
## lat                                   FALSE      FALSE
## lon                                   FALSE      FALSE
## Slope                                 FALSE      FALSE
## Solar.radiation                       FALSE      FALSE
## Next_Tmin                             FALSE      FALSE
## grouped_stationLow DEM station        FALSE      FALSE
## grouped_stationMedium DEM station     FALSE      FALSE
## grouped_stationHigh DEM station       FALSE      FALSE
## CC                                    FALSE      FALSE
## PPT                                   FALSE      FALSE
## 1 subsets of each size up to 16
## Selection Algorithm: exhaustive
##           RHmin RHmax Tmax_lapse Tmin_lapse WS  LH  lat lon Slope
## 1  ( 1 )  " "   " "   "*"        " "        " " " " " " " " " "
## 2  ( 1 )  " "   " "   "*"        " "        " " " " " " " " " "
## 3  ( 1 )  " "   " "   "*"        " "        " " " " " " " " " "
## 4  ( 1 )  " "   " "   "*"        " "        " " "*" " " " " " "
## 5  ( 1 )  " "   " "   "*"        " "        "*" "*" " " " " " "
## 6  ( 1 )  " "   " "   "*"        " "        "*" "*" " " " " " "
## 7  ( 1 )  " "   " "   "*"        " "        "*" "*" " " " " " "
## 8  ( 1 )  " "   " "   "*"        " "        "*" "*" " " "*" "*"
## 9  ( 1 )  " "   "*"   "*"        " "        "*" "*" " " "*" "*"
## 10  ( 1 ) "*"   "*"   "*"        "*"        "*" "*" " " "*" " "
## 11  ( 1 ) "*"   "*"   "*"        "*"        "*" "*" " " "*" "*"
## 12  ( 1 ) "*"   "*"   "*"        "*"        "*" "*" " " "*" "*"
## 13  ( 1 ) "*"   "*"   "*"        "*"        "*" "*" " " "*" "*"
## 14  ( 1 ) "*"   "*"   "*"        "*"        "*" "*" "*" "*" "*"
## 15  ( 1 ) "*"   "*"   "*"        "*"        "*" "*" "*" "*" "*"
## 16  ( 1 ) "*"   "*"   "*"        "*"        "*" "*" "*" "*" "*"
##           Solar.radiation Next_Tmin grouped_stationLow DEM station
## 1  ( 1 )  " "             " "       " "
## 2  ( 1 )  " "             "*"       " "
## 3  ( 1 )  " "             "*"       " "
## 4  ( 1 )  " "             "*"       " "
## 5  ( 1 )  " "             "*"       " "
## 6  ( 1 )  " "             "*"       " "
## 7  ( 1 )  " "             "*"       "*"
## 8  ( 1 )  " "             "*"       " "
## 9  ( 1 )  " "             "*"       " "
## 10  ( 1 ) " "             "*"       " "
## 11  ( 1 ) " "             "*"       " "
## 12  ( 1 ) " "             "*"       "*"
## 13  ( 1 ) "*"             "*"       "*"
## 14  ( 1 ) "*"             "*"       "*"
## 15  ( 1 ) "*"             "*"       "*"
## 16  ( 1 ) "*"             "*"       "*"
##           grouped_stationMedium DEM station grouped_stationHigh DEM station CC
## 1  ( 1 )  " "                               " "                             " "
## 2  ( 1 )  " "                               " "                             " "
## 3  ( 1 )  " "                               " "                             "*"
## 4  ( 1 )  " "                               " "                             "*"
## 5  ( 1 )  " "                               " "                             "*"
## 6  ( 1 )  " "                               " "                             "*"
## 7  ( 1 )  " "                               " "                             "*"
## 8  ( 1 )  " "                               " "                             "*"
## 9  ( 1 )  " "                               " "                             "*"
## 10  ( 1 ) " "                               " "                             "*"
## 11  ( 1 ) " "                               " "                             "*"
## 12  ( 1 ) " "                               " "                             "*"
## 13  ( 1 ) " "                               " "                             "*"
## 14  ( 1 ) " "                               " "                             "*"
## 15  ( 1 ) "*"                               " "                             "*"
## 16  ( 1 ) "*"                               "*"                             "*"
##           PPT
## 1  ( 1 )  " "
## 2  ( 1 )  " "
## 3  ( 1 )  " "
## 4  ( 1 )  " "
## 5  ( 1 )  " "
## 6  ( 1 )  "*"
## 7  ( 1 )  "*"
## 8  ( 1 )  "*"
## 9  ( 1 )  "*"
## 10  ( 1 ) "*"
## 11  ( 1 ) "*"
## 12  ( 1 ) "*"
## 13  ( 1 ) "*"
## 14  ( 1 ) "*"
## 15  ( 1 ) "*"
## 16  ( 1 ) "*"
```

For instance, if we consider the best model fitted with four predictors, we should look at the row corresponding to the fourth iteration: it indicates that the best fit with four covariates contains Tmax_lapse, LH, Next_Tmin and CC. The most relevant covariate, that is selected since the first iteration, is Tmax_lapse, while the last predictor to be added is the "High DEM station" level of the categorical variable "grouped_station".

1.1.2 Best model according to AIC, BIC, adjusted R^2 and Mallow's C_p

We plot four graphs to select the best model according to the following criteria: AIC, BIC, adjusted R^2 and Mallow's C_p.

```r
rss = summary$rss
n = nrow(data)
aic_values = n * log(rss / n) + 2 * (1:length(rss))

stats_matrix <- cbind(num_predictors = 1:length(summary$bic),
                      AIC = aic_values,
                      BIC = summary$bic,
                      Cp = summary$cp,
                      AdjR2 = summary$adjr2)

stats_df <- as.data.frame(stats_matrix)
plot_list <- list()


for (stat in colnames(stats_df)[-1]){
  if (stat != "AdjR2"){

  min_stats_row <- stats_df[which.min(stats_df[[stat]]), 1]
  plot = ggplot(stats_df, aes(x= num_predictors, y = .data[[stat]])) +
  geom_point()+
  geom_line(color="blue")+
  geom_vline(xintercept = min_stats_row, color = "red", linetype = "dashed") +
  labs(x = "Number of Predictors", y = stat, "Value") +
  theme_minimal()

  plot_list[[stat]] <- plot

  } else {
      max_stats_row <- stats_df[which.max(stats_df[[stat]]), 1]

  plot = ggplot(stats_df, aes(x= num_predictors, y = .data[[stat]])) +
  geom_point()+
  geom_line(color="blue")+
  geom_vline(xintercept = max_stats_row, color = "red", linetype = "dashed") +
  labs(x = "Number of Predictors", y = stat, "Value") +
  theme_minimal()

  plot_list[[stat]] <- plot
  }

}

grid.arrange(grobs = plot_list, ncol = 2)
```



The selection of the best overall model varies across different criteria and is not consistent, however this model typically consists of 12 to 14 covariates. It's worth mentioning that the improvement between successive models significantly decreases after the fifth iteration, suggesting that the differences among the models selected as the best are minimal.

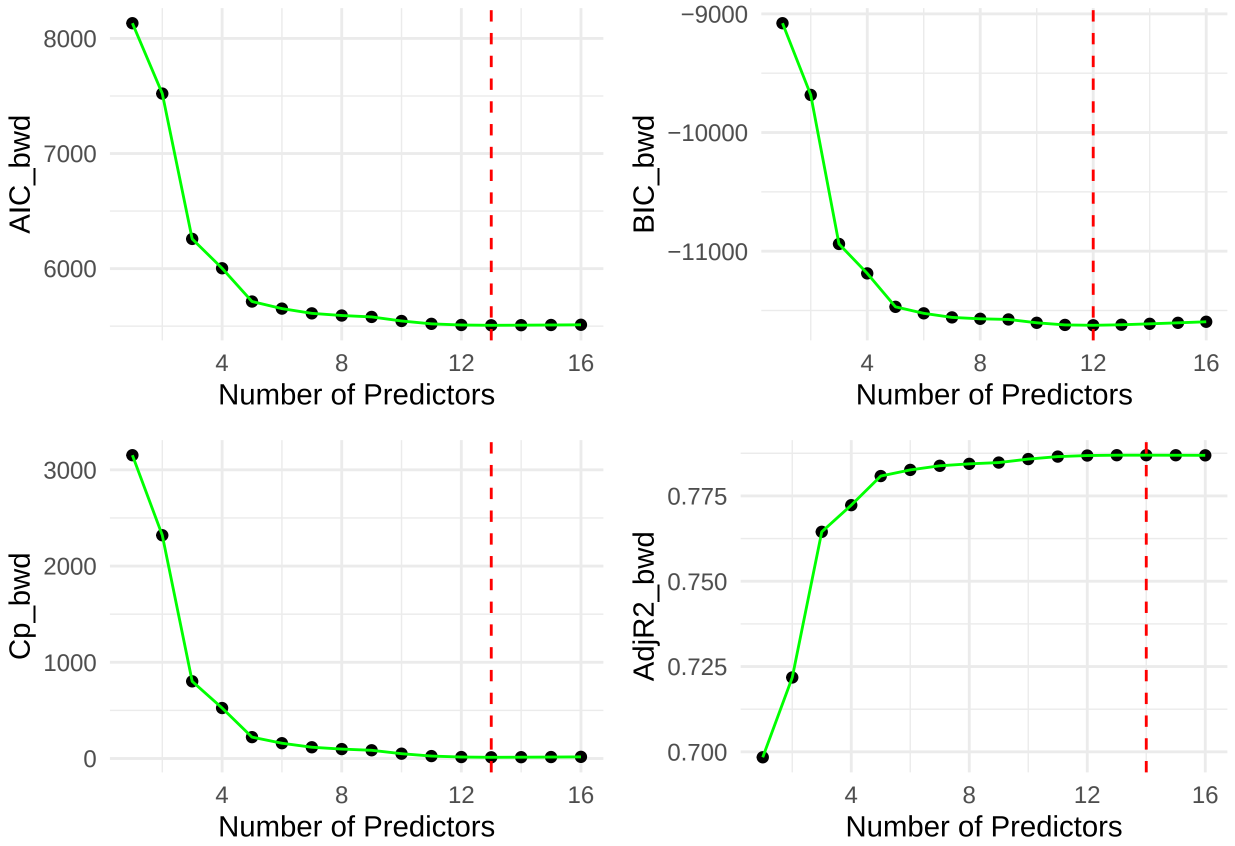
We fit the model with 13 predictors, best model according to both the AIC and C_p criteria:

```
#Create a dummy variable for the level "low DEM station" of the categorical variable

data$grouped_station_low = as.numeric(data$grouped_station == "Low DEM station")

ols_bs = lm (
  Next_Tmax  ~ RHmin + RHmax + Tmax_lapse + Tmin_lapse + WS + LH + lon + Slope +
    Solar.radiation + Next_Tmin + data$grouped_station_low + CC + PPT,
  data = data
)
summary(ols_bs)
```

```
##
## Call:
## lm(formula = Next_Tmax ~ RHmin + RHmax + Tmax_lapse + Tmin_lapse +
##     WS + LH + lon + Slope + Solar.radiation + Next_Tmin + data$grouped_station_low +
##     CC + PPT, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8121 -0.8286  0.0117  0.8488  6.2646
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.601e+02  2.891e+01   5.539 3.14e-08 ***
## RHmin                     1.300e-02  2.713e-03   4.790 1.70e-06 ***
## RHmax                    -2.097e-02  3.148e-03  -6.661 2.91e-11 ***
## Tmax_lapse                5.624e-01  1.514e-02  37.144  < 2e-16 ***
## Tmin_lapse               -1.145e-01  1.974e-02  -5.797 7.01e-09 ***
## WS                       -1.633e-01  8.357e-03 -19.538  < 2e-16 ***
## LH                        8.569e-03  5.694e-04  15.048  < 2e-16 ***
## lon                      -1.195e+00  2.279e-01  -5.243 1.63e-07 ***
## Slope                     5.606e-02  1.329e-02   4.219 2.49e-05 ***
## Solar.radiation           8.827e-05  4.159e-05   2.123 0.033819 *
## Next_Tmin                 4.495e-01  1.466e-02  30.656  < 2e-16 ***
## data$grouped_station_low -1.350e-01  3.993e-02  -3.381 0.000725 ***
## CC                       -3.752e+00  1.439e-01 -26.074  < 2e-16 ***
## PPT                       1.836e-01  2.167e-02   8.472  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.436 on 7574 degrees of freedom
## Multiple R-squared:  0.7873, Adjusted R-squared:  0.7869
## F-statistic:  2157 on 13 and 7574 DF,  p-value: < 2.2e-16
```

1.1.3 Backward selection

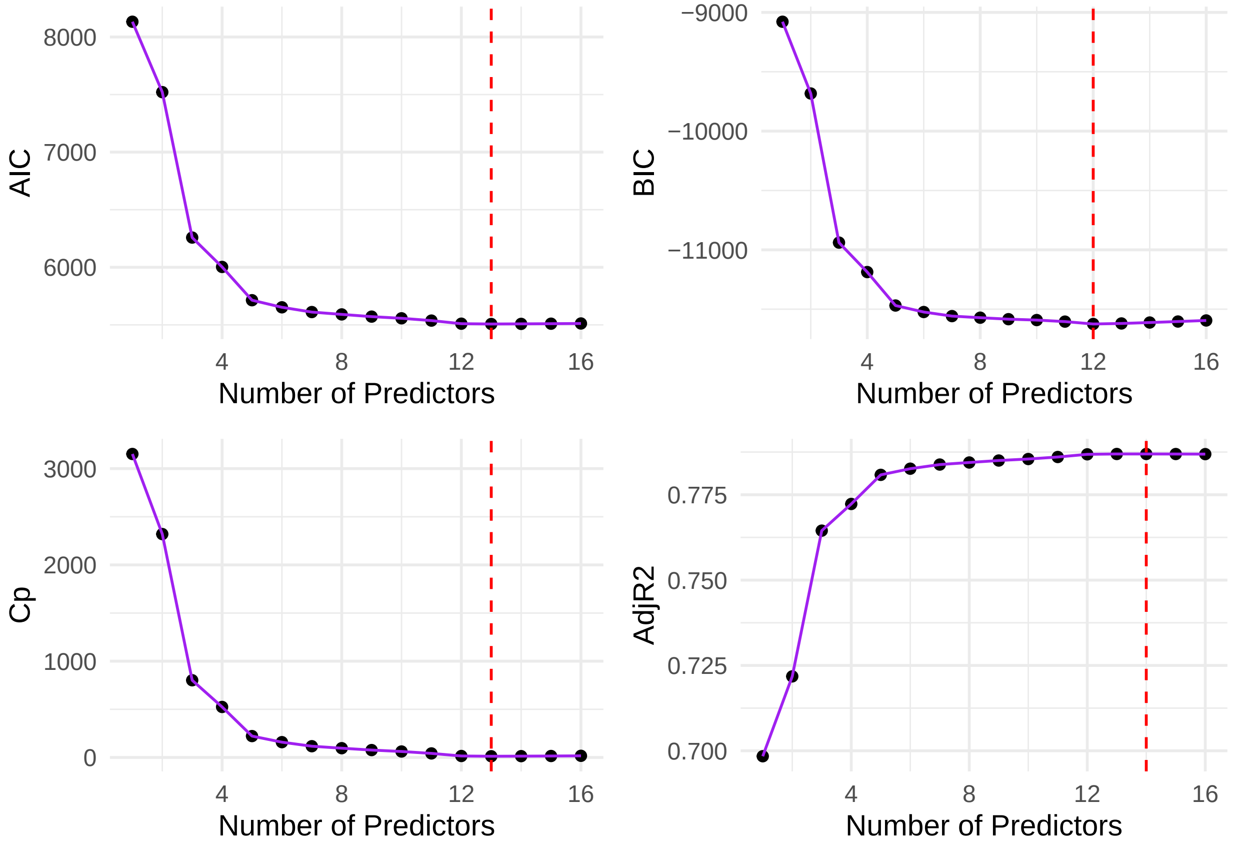
```
ols.bwd = regsubsets (
  Next_Tmax  ~ RHmin + RHmax + Tmax_lapse + Tmin_lapse + WS + LH + lat + lon +
    Slope + Solar.radiation + Next_Tmin + grouped_station + CC + PPT,
  data = data,
  nvmax = 16,
  method = "backward"
)
summ_bwd = summary(ols.bwd)
```



The result obtained with this method is the same as in the previous analysis, although the predictors selected for some iterations are different compared to the best subset selection.

1.1.4 Forward selection

```
ols.fwd = regsubsets (
  Next_Tmax  ~ RHmin + RHmax + Tmax_lapse + Tmin_lapse + WS + LH + lat + lon +
  Slope + Solar.radiation + Next_Tmin + grouped_station + CC + PPT,
  data = data,
  nvmax = 16,
  method = "forward"
)
summ_fwd = summary(ols.fwd)
```



The result obtained with this method is the same as before.

1.1.5 Validation set approach

We now want to use the validation set approach to compute the MSE of the model with 13 predictors previously selected. We use random sampling to create three different training sets (as big as 80% of the dataset) and validation sets (as big as 20% of the dataset). The results range between 2.96 and 3.08.

```
set.seed(123)

#Generate random indices for splitting
indices = sample(1:nrow(data), size = nrow(data), replace = FALSE)

#Proportion for splitting (80% train, 20% validation)
train_prop = 0.8
train_size = floor(train_prop * nrow(data))

train_data = data[indices[1:train_size], ]
validation_data = data[indices[(train_size + 1):nrow(data)], ]

model = rpart(Next_Tmax  ~ RHmin + RHmax + Tmax_lapse + Tmin_lapse + WS + LH + lon + Slope + Solar.radi

predictions = predict(model, newdata = validation_data)

mse = mean((predictions - validation_data$Next_Tmax)^2)
mse
```

```
## [1] 3.034966
```

```
set.seed(1)

#Generate random indices for splitting
indices = sample(1:nrow(data), size = nrow(data), replace = FALSE)

#Proportion for splitting (80% train, 20% validation)
train_prop = 0.8
train_size = floor(train_prop * nrow(data))

train_data = data[indices[1:train_size], ]
validation_data = data[indices[(train_size + 1):nrow(data)], ]

model = rpart(Next_Tmax  ~ RHmin + RHmax + Tmax_lapse + Tmin_lapse + WS + LH + lon + Slope + Solar.radia

predictions = predict(model, newdata = validation_data)

mse = mean((predictions - validation_data$Next_Tmax)^2)
mse
```

```
## [1] 3.078287
```

```
set.seed(2)

#Generate random indices for splitting
indices = sample(1:nrow(data), size = nrow(data), replace = FALSE)

#Proportion for splitting (80% train, 20% validation)
train_prop = 0.8
train_size = floor(train_prop * nrow(data))

train_data = data[indices[1:train_size], ]
validation_data = data[indices[(train_size + 1):nrow(data)], ]

model = rpart(Next_Tmax  ~ RHmin + RHmax + Tmax_lapse + Tmin_lapse + WS + LH + lon + Slope + Solar.radia

predictions = predict(model, newdata = validation_data)

mse = mean((predictions - validation_data$Next_Tmax)^2)
mse
```

```
## [1] 2.96046
```

1.1.6 LOOCV cross-validation approach

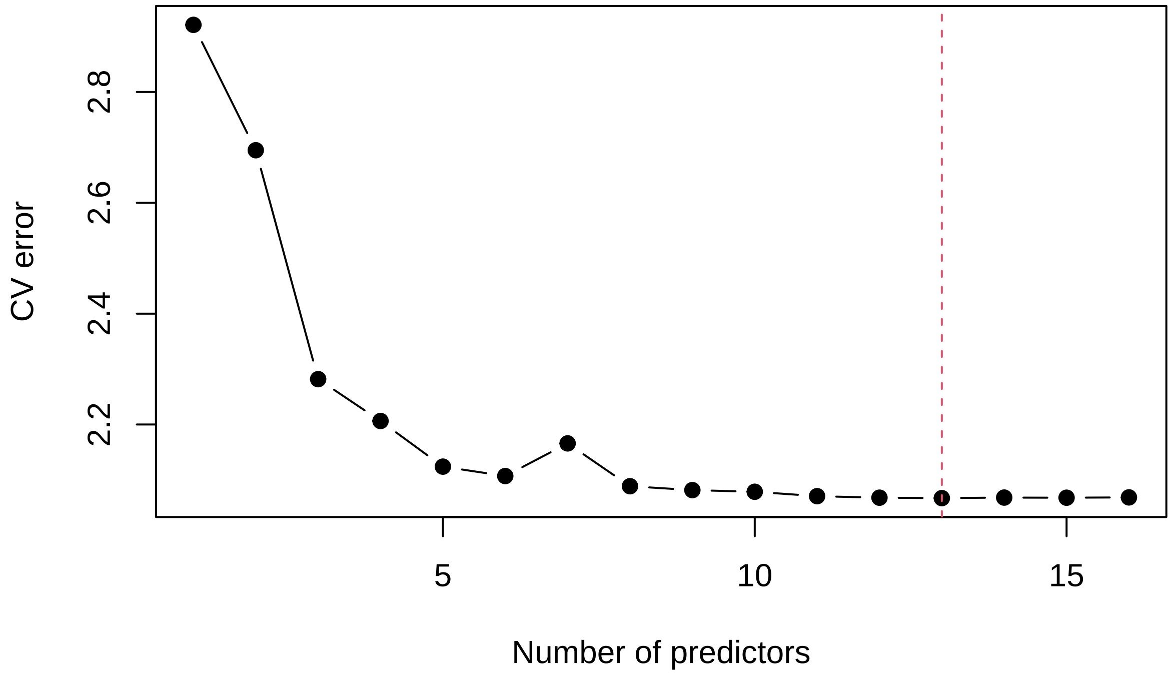
We use the leave-one-out cross-validation approach to select the best model according to the best subset selection. The lowest overall mean cross validation error is obtained with the best model with 13 predictors, hence the conclusions are the same as all the previous analysis.

```
p = 16
k = nrow(data)

set.seed (1234)
folds = sample (1:k, nrow(data), replace = F)
cv.errors = matrix (NA , k, p, dimnames = list(NULL , paste (1:p)))
```

```
for (j in 1:k) {
  best.fit = regsubsets (
    Next_Tmax  ~ RHmin + RHmax + Tmax_lapse + Tmin_lapse + WS + LH + lat + lon + Slope + Solar.radiatio
    data = data[folds != j,],
    nvmax = 16
  )
  for (i in 1:p) {
    mat <-
      model.matrix(as.formula(best.fit$call[[2]]), data[folds == j,])
    coefi <- coef(best.fit , id = i)
    xvars <- names(coefi)
    pred <- mat[, xvars] %*% coefi
    cv.errors[j, i] <- mean((data$Next_Tmax[folds == j] - pred) ^ 2)
  }
}

cv.mean = colMeans(cv.errors)
cv.mean
```

```
##        1        2        3        4        5        6        7        8
## 2.921073 2.694867 2.281833 2.206321 2.123977 2.107005 2.165910 2.088637
##        9       10       11       12       13       14       15       16
## 2.081607 2.078631 2.070682 2.067948 2.067188 2.068139 2.067884 2.068488
```

```
par(mfrow = c(1, 1))
plot(
  cv.mean ,
  type = "b",
  pch = 19,
  xlab = "Number of predictors",
  ylab = "CV error"
)
abline(v = which.min(cv.mean),
       col = 2,
       lty = 2)
```


CV error
2.2
2.4
2.6
2.8
5
10
15
Number of predictors