Generalized Additive Model

2024-05-07

MODEL SELECTION

We decided to apply best subset selection as our method of model selection as it considers all possible combinations of features while determining the set that would produce the highest performing model. As the number of predictors 'p' here is only 11, it is still a computationally feasible approach. We implemented the method on white and red wine datasets separately as the distribution of the outcome variable was slightly different - calling for separate approaches in how we predict the quality of each type. After summarizing the results of the best subset calculated at each possible number of predictors, we plotted curves to determine the subsets that produced lowest BIC values. Results from the white dataset showed that the most optimal number of predictors for any model in general was 8 and for the red wines - 6. This plots also helped us choose which predictors were of highest importance at these respective model sizes.

```
#install.packages('leaps')
library(leaps)
redwines <- read.csv('winequality-red.csv', sep = ';')
whitewines <- read.csv('winequality-white.csv', sep = ';')
regfit.full_white <- regsubsets(whitewines$quality ~ ., data = whitewines, nvmax = 12)
reg.summary_white <- summary(regfit.full_white)
(reg.summary_white)
```

```
## Subset selection object
## Call: regsubsets.formula(whitewines$quality ~ ., data = whitewines,
##     nvmax = 12)
## 11 Variables  (and intercept)
##                       Forced in Forced out
## fixed.acidity             FALSE      FALSE
## volatile.acidity          FALSE      FALSE
## citric.acid               FALSE      FALSE
## residual.sugar            FALSE      FALSE
## chlorides                 FALSE      FALSE
## free.sulfur.dioxide       FALSE      FALSE
## total.sulfur.dioxide      FALSE      FALSE
## density                   FALSE      FALSE
## pH                        FALSE      FALSE
## sulphates                 FALSE      FALSE
## alcohol                   FALSE      FALSE
## 1 subsets of each size up to 11
## Selection Algorithm: exhaustive
##           fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1  ( 1 )  " "           " "              " "         " "             " "
## 2  ( 1 )  " "           "*"              " "         " "             " "
## 3  ( 1 )  " "           "*"              " "         "*"             " "
## 4  ( 1 )  " "           "*"              " "         "*"             " "
## 5  ( 1 )  " "           "*"              " "         "*"             " "
## 6  ( 1 )  " "           "*"              " "         "*"             " "
## 7  ( 1 )  " "           "*"              " "         "*"             " "
```

```
## 8  ( 1 )  "*"           "*"               " "          "*"            " "
## 9  ( 1 )  "*"           "*"               " "          "*"            " "
## 10  ( 1 ) "*"           "*"               " "          "*"            "*"
## 11  ( 1 ) "*"           "*"               "*"          "*"            "*"
##           free.sulfur.dioxide total.sulfur.dioxide density pH  sulphates
## 1  ( 1 )  " "                 " "                 " "     " " " "
## 2  ( 1 )  " "                 " "                 " "     " " " "
## 3  ( 1 )  " "                 " "                 " "     " " " "
## 4  ( 1 )  "*"                 " "                 " "     " " " "
## 5  ( 1 )  " "                 " "                 "*"     "*" " "
## 6  ( 1 )  " "                 " "                 "*"     "*" "*"
## 7  ( 1 )  "*"                 " "                 "*"     "*" "*"
## 8  ( 1 )  "*"                 " "                 "*"     "*" "*"
## 9  ( 1 )  "*"                 "*"                 "*"     "*" "*"
## 10  ( 1 ) "*"                 "*"                 "*"     "*" "*"
## 11  ( 1 ) "*"                 "*"                 "*"     "*" "*"
##           alcohol
## 1  ( 1 )  "*"
## 2  ( 1 )  "*"
## 3  ( 1 )  "*"
## 4  ( 1 )  "*"
## 5  ( 1 )  "*"
## 6  ( 1 )  "*"
## 7  ( 1 )  "*"
## 8  ( 1 )  "*"
## 9  ( 1 )  "*"
## 10  ( 1 ) "*"
## 11  ( 1 ) "*"
```

```
names(reg.summary_white)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
(reg.summary_white$adjr2)
```

```
##  [1] 0.1895598 0.2399208 0.2580716 0.2633925 0.2703282 0.2757705 0.2790891
##  [8] 0.2805767 0.2805130 0.2803931 0.2802536
```

```
which.min(reg.summary_white$bic)
```

```
## [1] 8
```

```
regfit.full_red <- regsubsets(redwines$quality ~ ., data = redwines, nvmax = 12)
reg.summary_red <- summary(regfit.full_red)
(reg.summary_red)
```

```
## Subset selection object
## Call: regsubsets.formula(redwines$quality ~ ., data = redwines, nvmax = 12)
## 11 Variables  (and intercept)
##                      Forced in Forced out
## fixed.acidity            FALSE      FALSE
## volatile.acidity         FALSE      FALSE
## citric.acid              FALSE      FALSE
## residual.sugar           FALSE      FALSE
## chlorides                FALSE      FALSE
## free.sulfur.dioxide      FALSE      FALSE
## total.sulfur.dioxide     FALSE      FALSE
```

```
## density                 FALSE      FALSE
## pH                      FALSE      FALSE
## sulphates               FALSE      FALSE
## alcohol                 FALSE      FALSE
## 1 subsets of each size up to 11
## Selection Algorithm: exhaustive
##           fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1  ( 1 )  " "           " "              " "         " "            " "
## 2  ( 1 )  " "           "*"              " "         " "            " "
## 3  ( 1 )  " "           "*"              " "         " "            " "
## 4  ( 1 )  " "           "*"              " "         " "            " "
## 5  ( 1 )  " "           "*"              " "         " "            "*"
## 6  ( 1 )  " "           "*"              " "         " "            "*"
## 7  ( 1 )  " "           "*"              " "         " "            "*"
## 8  ( 1 )  " "           "*"              "*"         " "            "*"
## 9  ( 1 )  " "           "*"              "*"         "*"            "*"
## 10  ( 1 ) "*"           "*"              "*"         "*"            "*"
## 11  ( 1 ) "*"           "*"              "*"         "*"            "*"
##           free.sulfur.dioxide total.sulfur.dioxide density pH  sulphates
## 1  ( 1 )  " "                 " "                 " "     " " " "
## 2  ( 1 )  " "                 " "                 " "     " " " "
## 3  ( 1 )  " "                 " "                 " "     " " "*"
## 4  ( 1 )  " "                 "*"                 " "     " " "*"
## 5  ( 1 )  " "                 "*"                 " "     " " "*"
## 6  ( 1 )  " "                 "*"                 " "     "*" "*"
## 7  ( 1 )  "*"                 "*"                 " "     "*" "*"
## 8  ( 1 )  "*"                 "*"                 " "     "*" "*"
## 9  ( 1 )  "*"                 "*"                 " "     "*" "*"
## 10  ( 1 ) "*"                 "*"                 " "     "*" "*"
## 11  ( 1 ) "*"                 "*"                 "*"     "*" "*"
##           alcohol
## 1  ( 1 )  "*"
## 2  ( 1 )  "*"
## 3  ( 1 )  "*"
## 4  ( 1 )  "*"
## 5  ( 1 )  "*"
## 6  ( 1 )  "*"
## 7  ( 1 )  "*"
## 8  ( 1 )  "*"
## 9  ( 1 )  "*"
## 10  ( 1 ) "*"
## 11  ( 1 ) "*"
```

```
names(reg.summary_red)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
(reg.summary_red$adjr2)
```

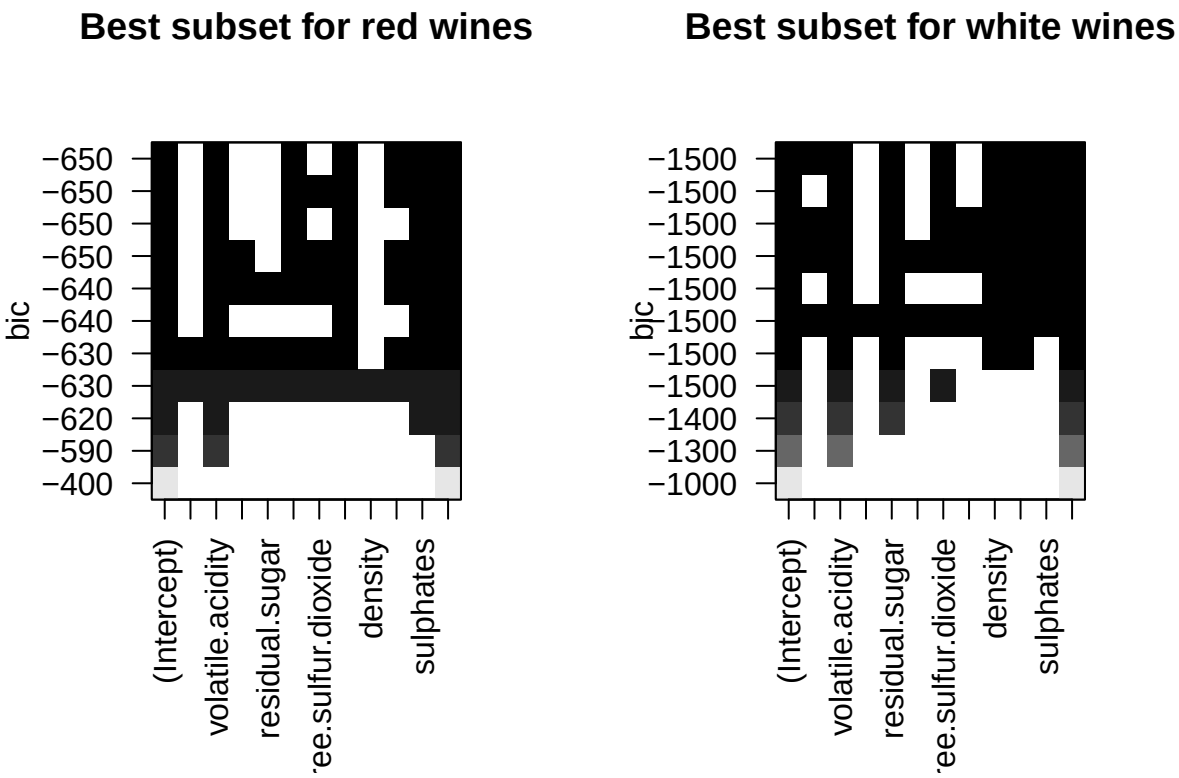
```
##  [1] 0.2262502 0.3161465 0.3346482 0.3421357 0.3494588 0.3547509 0.3566527
##  [8] 0.3567060 0.3565489 0.3562479 0.3561195
```

```
which.min(reg.summary_red$bic)
```

```
## [1] 6
```

```
par(mfrow = c(1,2))

plot(regfit.full_red, scale = "bic", main = 'Best subset for red wines')
plot(regfit.full_white, scale = "bic", main = 'Best subset for white wines')
```



Once we determined the best subset of predictors to train the model on, we chose to use a Generalized Additive Model to explore the possibility that each feature followed varying true models, and should thus be approached using an additive technique. This allows us to apply non-linear relationships to the data with additional flexibility of each predictor's contributions being considered separately The smoothing parameter here is automatically tuned using the gam(), which uses a technique known as backfitting. Therefore, there was no need for manual tuning of hyperparameters in this case.

We see that the model outputs a 10-fold cross validation RMSE of 0.7267, an R-squared of 0.3355, and MAE of 0.5697 using the most optimal model where select = TRUE. This hyperparameter was automatically determined, and signifies that the most optimal of the 2 possible GAM models applies additional penalties on the model curve to spaces where the effect of splining is null. These regions are known as null spaces, and the TRUE option likely uses less effective degrees of freedom than its FALSE counterpart.

When applied separately using the previously mentioned best subset for the red wines data, we obtain an RMSE of 0.6397, an R-squared of 0.3754, and MAE of 0.4966, which are not significant improvements from the white wine's GAM model.

```
library(mgcv)
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.9-0. For overview type 'help("mgcv-package")'.
```

```r
library(boot)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'lattice'
```

```
## The following object is masked from 'package:boot':
##
##     melanoma
```

```r
set.seed(1)
ctrl <- trainControl(method = "cv")

model_white <- train(quality ~ fixed.acidity + volatile.acidity + residual.sugar + free.sulfur.dioxide +

model_white
```

```
## Generalized Additive Model using Splines
##
## 4898 samples
##    8 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 4409, 4408, 4408, 4408, 4408, 4408, ...
## Resampling results across tuning parameters:
##
##   select  RMSE       Rsquared   MAE
##   FALSE   0.7228000  0.3377613  0.5678348
##    TRUE   0.7276612  0.3362123  0.5687308
##
## Tuning parameter 'method' was held constant at a value of GCV.Cp
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were select = FALSE and method = GCV.Cp.
```

```r
#model_gam <- gam(quality ~ s(volatile.acidity) + s(total.sulfur.dioxide) + s(chlorides) + s(pH) + s(su

#plot(model_gam)

model_red <- train(quality ~ volatile.acidity + total.sulfur.dioxide + chlorides + pH + sulphates + alc

model_red
```

```
## Generalized Additive Model using Splines
##
## 1599 samples
##    6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1440, 1439, 1439, 1439, 1439, 1439, ...
## Resampling results across tuning parameters:
##
```

```
##   select  RMSE       Rsquared   MAE
##   FALSE   0.6379709  0.3794564  0.4952104
##    TRUE   0.6357140  0.3827440  0.4945756
##
## Tuning parameter 'method' was held constant at a value of GCV.Cp
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were select = TRUE and method = GCV.Cp.
```