final_project

2024-05-04

```
library(leaps)

set.seed(1)

red_wine <- read.csv("winequality-red.csv", sep = ";")
white_wine <- read.csv("winequality-white.csv", sep = ";")

# Perform best subset selection for white wines
regfit.full_white <- regsubsets(quality ~ ., data = white_wine, nvmax = 12)
reg.summary_white <- summary(regfit.full_white)

(reg.summary_white)
```

```
## Subset selection object
## Call: regsubsets.formula(quality ~ ., data = white_wine, 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 ) "*"
```

```
(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
```

```
# Find best subset based on BIC
best_subset_white <- which.min(reg.summary_white$bic)

# Perform best subset selection for red wines
regfit.full_red <- regsubsets(quality ~ ., data = red_wine, nvmax = 12)
reg.summary_red <- summary(regfit.full_red)

(reg.summary_red)
```

```
## Subset selection object
## Call: regsubsets.formula(quality ~ ., data = red_wine, 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 )  " "           " "              " "         " "             " "
```

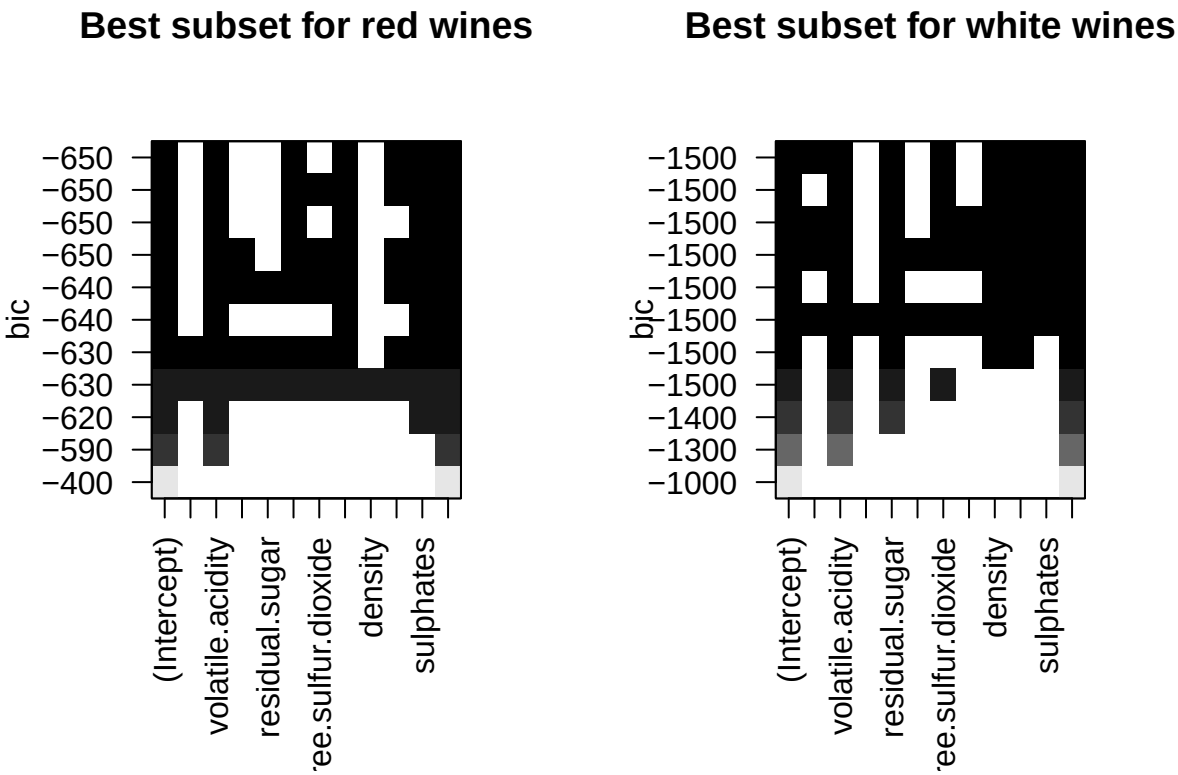
```
## 2  ( 1 )  " "           "*"             " "          " "           " "
## 3  ( 1 )  " "           "*"             " "          " "           " "
## 4  ( 1 )  " "           "*"             " "          " "           " "
## 5  ( 1 )  " "           "*"             " "          " "           "*"
## 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 ) "*"
```

```
(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
```

```
# Find best subset based on BIC
best_subset_red <- which.min(reg.summary_red$bic)

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')
```




```
library(caret)
```

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

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

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

```
set.seed(1)

# Define the range of k values
kset <- c(1:9, seq(10, 60, 5))

# Initialize vectors to store performance metrics
mse_values <- numeric(length(kset))
adj_r2_values <- numeric(length(kset))

for (i in seq_along(kset)) {
  # Train KNN model using LOOCV
  ctrl <- trainControl(method = "LOOCV")
  model_white <- train(quality ~ fixed.acidity +
                         volatile.acidity + residual.sugar
                       + free.sulfur.dioxide + density + pH +
                         sulphates + alcohol,
                       data = white_wine,
                       method = "knn",
                       tuneGrid = data.frame(k = kset[i]),
                       trControl = ctrl)
```

```
  # Get model performance metrics
  mse_values[i] <- model_white$results$RMSE
  # Can't use AIC since the AIC calculation is not directly applicable to the KNN model
  # Calculate R-squared manually
  R2 <- 1 - model_white$results$RMSE^2 / var(white_wine$quality)

  # Calculate adjusted R-squared manually
  n <- nrow(white_wine)
  k <- length(coef(model_white$finalModel))
  adj_r2_values[i] <- 1 - ((1 - R2) * (n - 1) / (n - k - 1))
}

# Find the index of the minimum MSE and maximum adjusted R-squared values
best_mse_index <- which.min(mse_values)
best_adj_r2_index <- which.max(adj_r2_values)

# Get the best performing values of k
best_mse_k <- kset[best_mse_index]
best_adj_r2_k <- kset[best_adj_r2_index]

# Print the MSE and adjusted R-squared values for the best performing k
cat("Best MSE value for white wine (k =", best_mse_k, "):",
    mse_values[best_mse_index], "\n")
```

```
## Best MSE value for white wine (k = 9 ): 0.7649898
```

```
cat("Best adjusted R^2 value for white wine (k =", best_adj_r2_k, "):",
    adj_r2_values[best_adj_r2_index], "\n")
```

```
## Best adjusted R^2 value for white wine (k = 9 ): 0.253898
```

```
library(caret)

set.seed(1)

# Define the range of k values
kset <- c(1:9, seq(10, 60, 5))

# Initialize vectors to store performance metrics
mse_values <- numeric(length(kset))
adj_r2_values <- numeric(length(kset))

for (i in seq_along(kset)) {
  # Train KNN model using LOOCV
  ctrl <- trainControl(method = "LOOCV")
  model_red <- train(quality ~ volatile.acidity +
                       citric.acid + chlorides + free.sulfur.dioxide +
                       total.sulfur.dioxide + pH +
                       sulphates + alcohol,
                       data = red_wine,
                       method = "knn",
                       tuneGrid = data.frame(k = kset[i]),
                       trControl = ctrl)
```

```
  # Get model performance metrics
  mse_values[i] <- model_red$results$RMSE
  # Can't use AIC since the AIC calculation is not directly applicable to the KNN model
  # Calculate R-squared manually
  R2 <- 1 - model_red$results$RMSE^2 / var(red_wine$quality)

  # Calculate adjusted R-squared manually
  n <- nrow(red_wine)
  k <- length(coef(model_red$finalModel))
  adj_r2_values[i] <- 1 - ((1 - R2) * (n - 1) / (n - k - 1))
}

# Find the index of the minimum MSE and maximum adjusted R-squared values
best_mse_index <- which.min(mse_values)
best_adj_r2_index <- which.max(adj_r2_values)

# Get the best performing values of k
best_mse_k <- kset[best_mse_index]
best_adj_r2_k <- kset[best_adj_r2_index]

# Print the MSE and adjusted R-squared values for the best performing k
cat("Best MSE value for red wine (k =", best_mse_k, "):",
    mse_values[best_mse_index], "\n")
```

```
## Best MSE value for red wine (k = 15 ): 0.7356221
```

```
cat("Best adjusted R^2 value for red wine (k =", best_adj_r2_k, "):",
    adj_r2_values[best_adj_r2_index], "\n")
```

```
## Best adjusted R^2 value for red wine (k = 15 ): 0.1702452
```