Homework 5

Zhezheng Jin

Contents

```
library(tidyverse)
library(caret)
library(mlbench)
library(pROC)
library(pdp)
library(ISLR)
library(caret)
library(AppliedPredictiveModeling)
library(tidymodels)
library(factoextra)
library(e1071)
```

1.Auto Data

```
# Data Import
auto <- read_csv("auto.csv") %>%
  mutate(
    mpg_cat = factor(mpg_cat, levels = c("low", "high")),
    origin = factor(origin, levels = 1:3),
    cylinders = as.factor(cylinders))
```

```
## Rows: 392 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): mpg_cat
## dbl (7): cylinders, displacement, horsepower, weight, acceleration, year, or...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
skimr::skim(auto)
```

Table 1: Data summary

Name	auto
Number of rows	392
Number of columns	8
Column type frequency:	
factor	3
numeric	5
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
cylinders	0	1	FALSE	5	4: 199, 8: 103, 6: 83, 3: 4
origin	0	1	FALSE	3	1: 245, 3: 79, 2: 68
mpg_cat	0	1	FALSE	2	low: 196, hig: 196

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
displacement	0	1	194.41	104.64	68	105.00	151.0	275.75	455.0	
horsepower	0	1	104.47	38.49	46	75.00	93.5	126.00	230.0	
weight	0	1	2977.58	849.40	1613	2225.25	2803.5	3614.75	5140.0	
acceleration	0	1	15.54	2.76	8	13.78	15.5	17.02	24.8	
year	0	1	75.98	3.68	70	73.00	76.0	79.00	82.0	

```
contrasts(auto$mpg_cat)
```

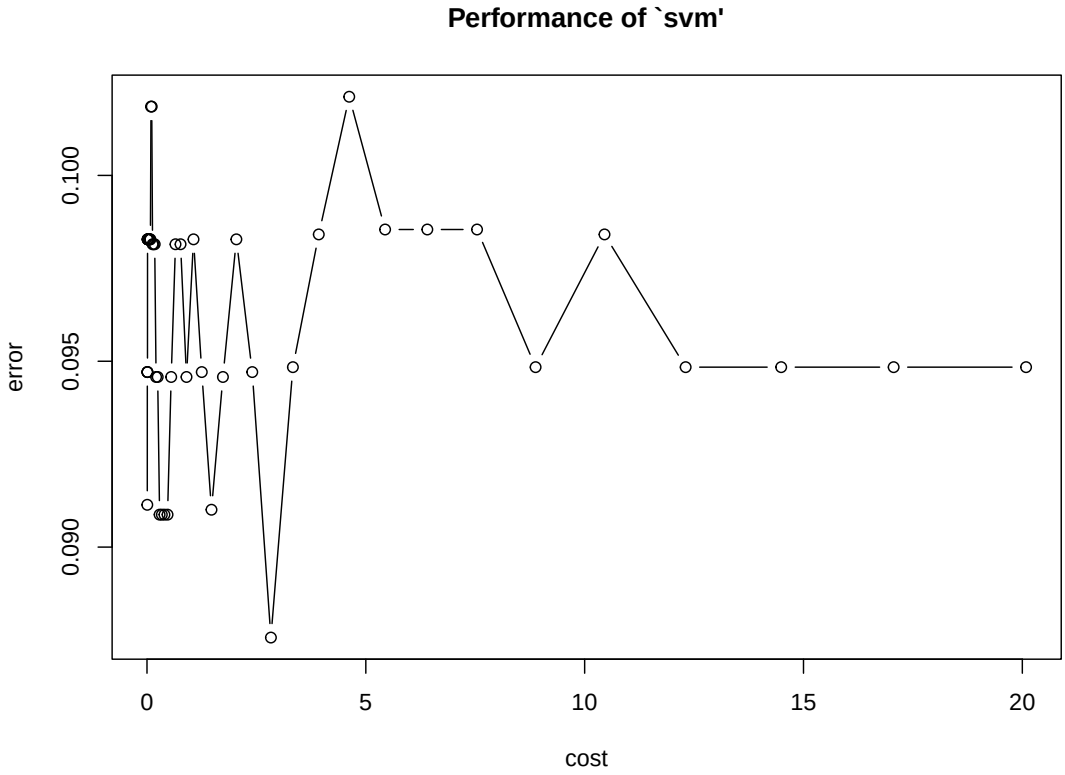
```
##      high
## low     0
## high    1
```

```
# data partition
set.seed(2358)
data_split <- initial_split(auto, prop = 0.7)
train <- training(data_split)
test <- testing(data_split)
```

The "auto" dataset contains 8 variables and 392 observations.

a. svm (linear kernel)

```
# use e1071
set.seed(23)
linear.tune <- tune.svm(mpg_cat ~ . ,
                        data = train,
                        kernel = "linear",
                        cost = exp(seq(-5,3,len=50)),
                        scale = TRUE)
plot(linear.tune)
```




```
linear.tune$best.parameters
```

```
##        cost
## 38 2.831528
```

```
best.linear <- linear.tune$best.model
summary(best.linear)
```

```
##
## Call:
## best.svm(x = mpg_cat ~ ., data = train, cost = exp(seq(-5, 3, len = 50)),
##     kernel = "linear", scale = TRUE)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  2.831528
##
## Number of Support Vectors:  52
##
##  ( 26 26 )
##
##
## Number of Classes:  2
```

```
##
## Levels:
##  low high
```

```
# Training error
confusionMatrix(data = linear.tune$best.model$fitted,
                reference = train$mpg_cat)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction low high
##       low  132   11
##       high   9  122
##
##                Accuracy : 0.927
##                  95% CI : (0.8895, 0.9548)
##     No Information Rate : 0.5146
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.8538
##
##  Mcnemar's Test P-Value : 0.8231
##
##             Sensitivity : 0.9362
##             Specificity : 0.9173
##          Pos Pred Value : 0.9231
##          Neg Pred Value : 0.9313
##              Prevalence : 0.5146
##          Detection Rate : 0.4818
##    Detection Prevalence : 0.5219
##       Balanced Accuracy : 0.9267
##
##        'Positive' Class : low
##
```

```
# Test error
pred.linear <- predict(best.linear, newdata = test)
confusionMatrix(data = pred.linear,
                reference = test$mpg_cat)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction low high
##       low   48    6
##       high   7   57
##
##                Accuracy : 0.8898
##                  95% CI : (0.819, 0.94)
##     No Information Rate : 0.5339
##     P-Value [Acc > NIR] : <2e-16
##
```

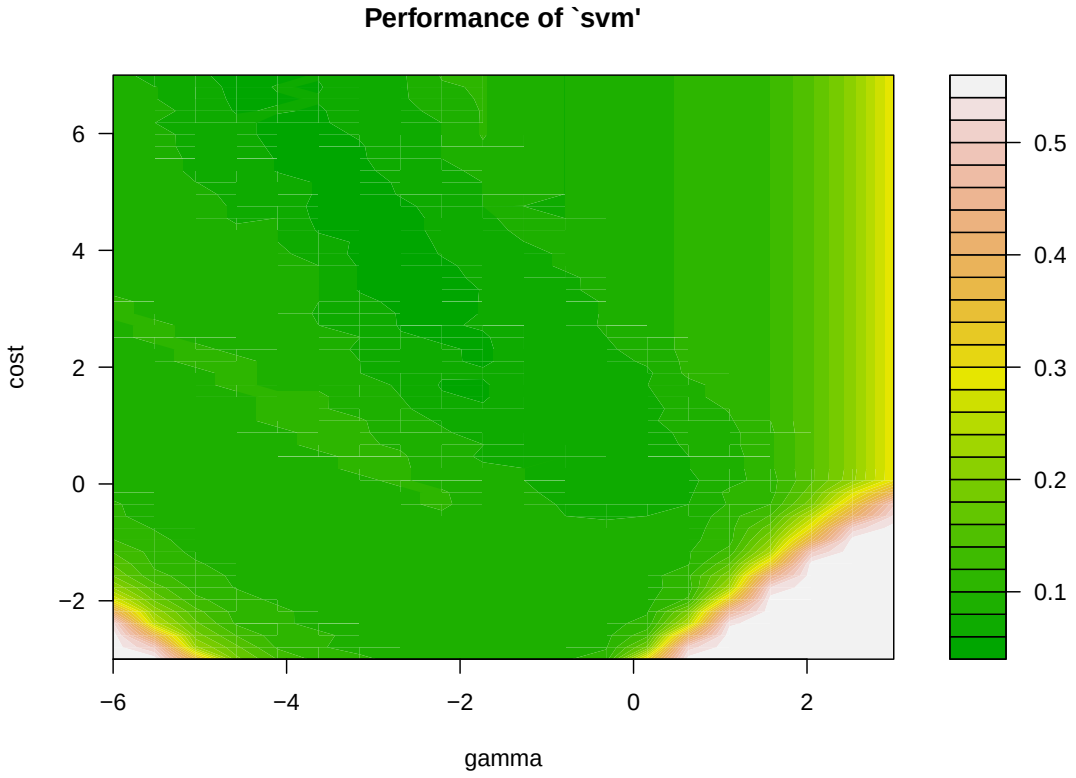
```
##                   Kappa : 0.7784
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8727
##             Specificity : 0.9048
##          Pos Pred Value : 0.8889
##          Neg Pred Value : 0.8906
##              Prevalence : 0.4661
##          Detection Rate : 0.4068
##    Detection Prevalence : 0.4576
##       Balanced Accuracy : 0.8887
##
##        'Positive' Class : low
##
```

In the training data, the support vector classifier (linear kernel) achieves an error rate of 7.3%. When applied to the test data, it achieves an error rate of 11.02%.

b. svm (radial kernel)

```
# use e1071
set.seed(23)
radial.tune <- tune.svm(mpg_cat ~ . ,
                        data = train,
                        kernel = "radial",
                        cost = exp(seq(-3,7,len=50)),
                        gamma = exp(seq(-6,3,len=20)))

plot(radial.tune, transform.y = log, transform.x = log,
     color.palette = terrain.colors)
```




```
radial.tune$best.parameters
```

```
##         gamma     cost
## 629 0.1096429 27.84158
```

```
best.radial <- radial.tune$best.model
summary(best.radial)
```

```
##
## Call:
## best.svm(x = mpg_cat ~ ., data = train, gamma = exp(seq(-6, 3, len = 20)),
##     cost = exp(seq(-3, 7, len = 50)), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  27.84158
##
## Number of Support Vectors:  55
##
##  ( 28 27 )
##
##
## Number of Classes:  2
```

```
##
## Levels:
##  low high
```

```
# Training error
confusionMatrix(data = radial.tune$best.model$fitted,
                reference = train$mpg_cat)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction low high
##       low  136    4
##       high   5  129
##
##                Accuracy : 0.9672
##                  95% CI : (0.9386, 0.9849)
##     No Information Rate : 0.5146
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9343
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.9645
##             Specificity : 0.9699
##          Pos Pred Value : 0.9714
##          Neg Pred Value : 0.9627
##              Prevalence : 0.5146
##          Detection Rate : 0.4964
##    Detection Prevalence : 0.5109
##       Balanced Accuracy : 0.9672
##
##        'Positive' Class : low
##
```

```
# Test error
pred.radial <- predict(best.radial, newdata = test)
confusionMatrix(data = pred.radial,
                reference = test$mpg_cat)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction low high
##       low   48    3
##       high   7   60
##
##                Accuracy : 0.9153
##                  95% CI : (0.8497, 0.9586)
##     No Information Rate : 0.5339
##     P-Value [Acc > NIR] : <2e-16
##
```

```
##                  Kappa : 0.8289
##
##  Mcnemar's Test P-Value : 0.3428
##
##             Sensitivity : 0.8727
##             Specificity : 0.9524
##          Pos Pred Value : 0.9412
##          Neg Pred Value : 0.8955
##              Prevalence : 0.4661
##          Detection Rate : 0.4068
##    Detection Prevalence : 0.4322
##       Balanced Accuracy : 0.9126
##
##        'Positive' Class : low
##
```

In the training data, the support vector machine with radial kernel achieves an error rate of 3.28%. When applied to the test data, it achieves an error rate of 8.47%.

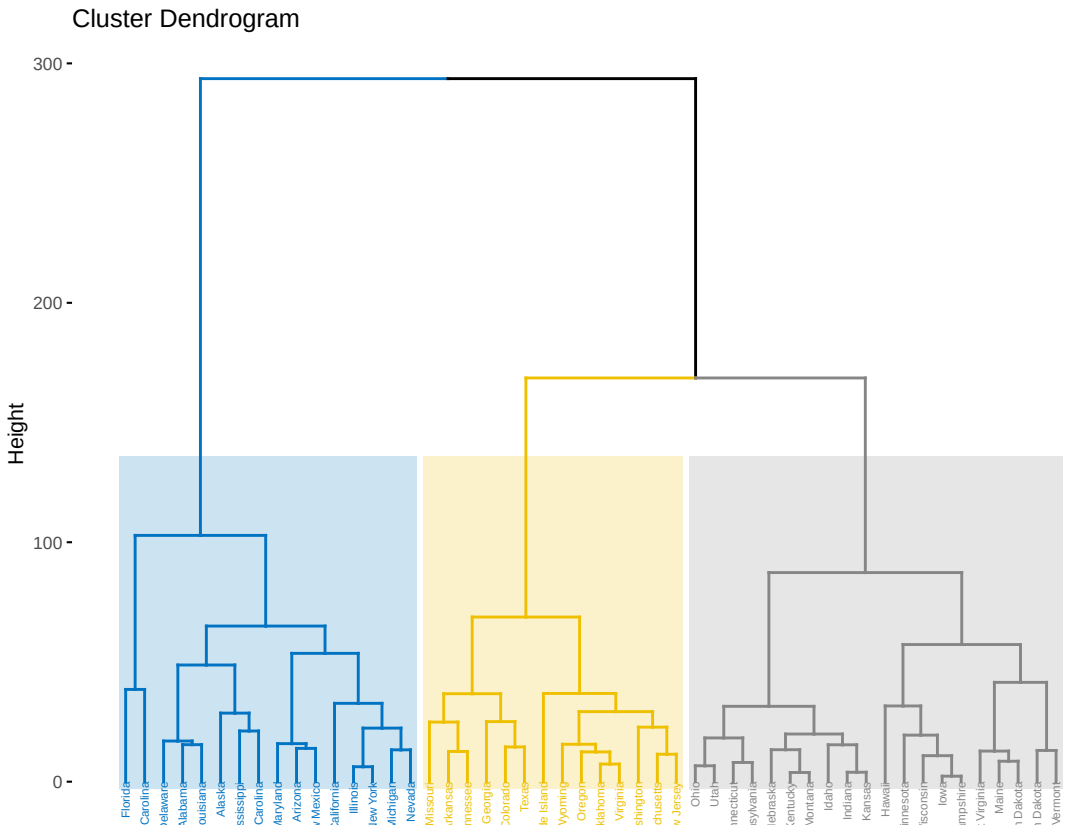
2. USArrests Data

```
# import data
data(USArrests)
```

a. Hierarchical Clustering

```
# using Complete linkage and Euclidean distance to cluster the states
hc.complete <- hclust(dist(USArrests), method = "complete")

# Cut the dendrogram at a height that results in three distinct clusters
fviz_dend(hc.complete, k = 3,
          cex = 0.4,
          palette = "jco",
          color_labels_by_k = TRUE,
          rect = TRUE, rect_fill = TRUE, rect_border = "jco",
          labels_track_height = 2.5)
```




```
ind3.complete <- cutree(hc.complete, 3)

# states belonging in clusters
USArrests[ind3.complete == 1,0] %>% t()
```

```
##       Alabama Alaska Arizona California Delaware Florida Illinois Louisiana
##       Maryland Michigan Mississippi Nevada New Mexico New York North Carolina
##       South Carolina
```

```
USArrests[ind3.complete == 2,0] %>% t()
```

```
##       Arkansas Colorado Georgia Massachusetts Missouri New Jersey Oklahoma
##       Oregon Rhode Island Tennessee Texas Virginia Washington Wyoming
```

```
USArrests[ind3.complete == 3,0] %>% t()
```

```
##       Connecticut Hawaii Idaho Indiana Iowa Kansas Kentucky Maine Minnesota
##       Montana Nebraska New Hampshire North Dakota Ohio Pennsylvania South Dakota
##       Utah Vermont West Virginia Wisconsin
```

The first cluster contains Alabama, Alaska, Arizona, California, Delaware, Florida, Illinois, Louisiana, Maryland, Michigan, Mississippi, Nevada, New Mexico, New York, North Carolina, South Carolina;

The second cluster contains Arkansas, Colorado, Georgia, Massachusetts, Missouri, New Jersey, Oklahoma, Oregon, Rhode Island, Tennessee, Texas, Virginia, Washington, Wyoming;

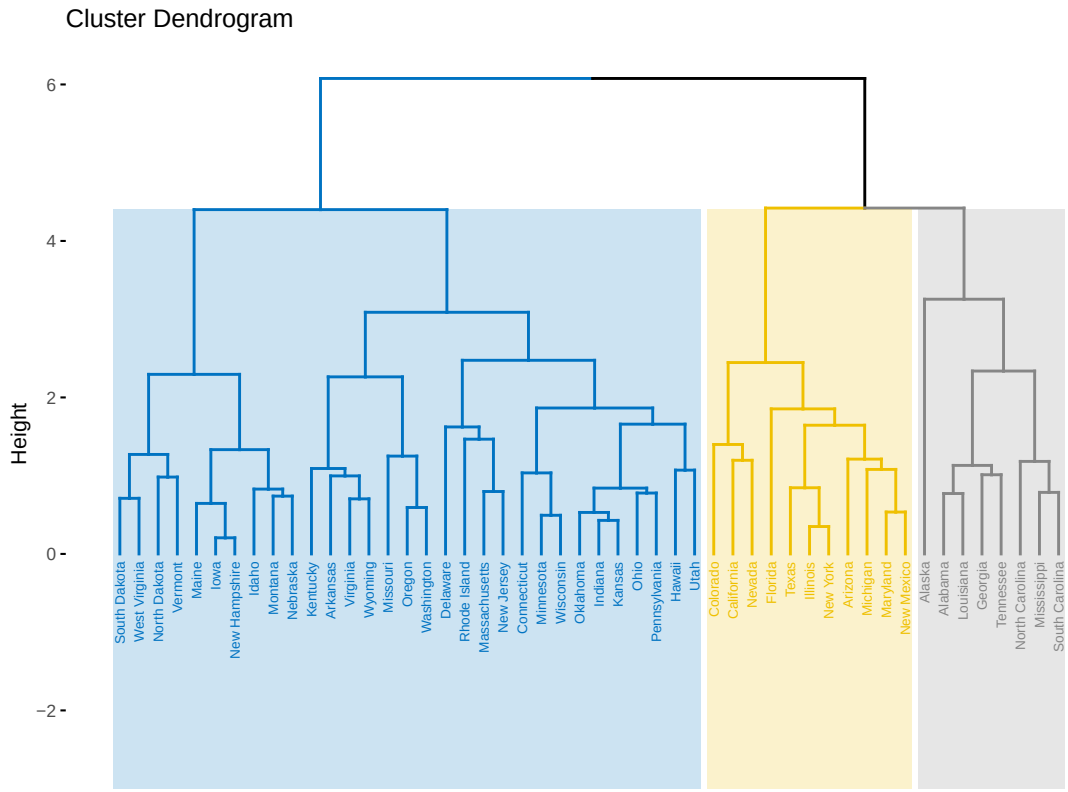
The third cluster contains Connecticut, Hawaii, Idaho, Indiana, Iowa, Kansas, Kentucky, Maine, Minnesota, Montana, Nebraska, New Hampshire, North Dakota, Ohio, Pennsylvania, South Dakota, Utah, Vermont, West Virginia, Wisconsin.

b. Hierarchical Clustering after scaling

```
# scale data
USArrests_scaled <- scale(USArrests, center = TRUE, scale = TRUE)

# Using Complete linkage and Euclidean distance to cluster the states
hc.complete.scaled <- hclust(dist(USArrests_scaled), method = "complete")

# Cut the dendrogram at a height that results in three distinct clusters
fviz_dend(hc.complete.scaled, k = 3,
          cex = 0.5,
          palette = "jco",
          color_labels_by_k = TRUE,
          rect = TRUE, rect_fill = TRUE, rect_border = "jco",
          labels_track_height = 2.5)
```




```
ind3.complete.scaled <- cutree(hc.complete.scaled, 3)

# states belonging in clusters
USArrests_scaled[ind3.complete.scaled == 1,0] %>% t()
```

```
##       Alabama Alaska Georgia Louisiana Mississippi North Carolina South Carolina
##       Tennessee
```

```
USArrests_scaled[ind3.complete.scaled == 2,0] %>% t()
```

```
##       Arizona California Colorado Florida Illinois Maryland Michigan Nevada
##       New Mexico New York Texas
```

```
USArrests_scaled[ind3.complete.scaled == 3,0] %>% t()
```

```
##       Arkansas Connecticut Delaware Hawaii Idaho Indiana Iowa Kansas Kentucky
##       Maine Massachusetts Minnesota Missouri Montana Nebraska New Hampshire
##       New Jersey North Dakota Ohio Oklahoma Oregon Pennsylvania Rhode Island
##       South Dakota Utah Vermont Virginia Washington West Virginia Wisconsin
##       Wyoming
```

The first cluster contains Alabama, Alaska, Georgia, Louisiana, Mississippi, North Carolina, South Carolina, Tennessee;

The second cluster contains Arizona, California, Colorado, Florida, Illinois, Maryland, Michigan, Nevada, New Mexico, New York, Texas;

The third cluster contains Arkansas, Connecticut, Delaware, Hawaii, Idaho, Indiana, Iowa, Kansas, Kentucky, Maine, Massachusetts, Minnesota, Missouri, Montana, Nebraska, New Hampshire, New Jersey, North Dakota, Ohio, Oklahoma, Oregon, Pennsylvania, Rhode Island, South Dakota, Utah, Vermont, Virginia, Washington, West Virginia, Wisconsin, Wyoming.

c. Differences between with or without scaling

Based on the results from 2a and 2b, scaling the variables changed the clustering results. This difference is attributed to how hierarchical clustering computes the distances between observations. Specifically, it uses the Euclidean distance, which is sensitive to the scales of the variables involved. When variables have different scales or units, those with larger magnitudes—such as `assault` or `urbanpop` in our case—may disproportionately influence the clustering, comparing to `murder` or `rape`. This can result in a bias where the clustering is dominated by one or two variables. Consequently, to prevent any single variable from overshadowing others and to ensure a more balanced contribution from all variables, it is advisable to scale the variables before performing distance calculations in the clustering process.

In my opinion, it is usually beneficial to scale the variables before calculating the inter-observation dissimilarities in hierarchical clustering. This approach helps to minimize bias resulting from variations in variable scales, ensuring that each variable contributes equally to the clustering process. However, there are exceptions where scaling might not be necessary or could even be inappropriate, depending on the specific situation of the data.