IIMT2641 Assignment 5

Sibo Ding

Spring 2023

# Q1 Tree Models

## Load the Data

state <- read.csv("StateData.csv")  
head(state) # First 6 rows

## Population Income Illiteracy LifeExp Murder HighSchoolGrad Frost Area  
## 1 3615 3624 2.1 69.05 15.1 41.3 20 50708  
## 2 365 6315 1.5 69.31 11.3 66.7 152 566432  
## 3 2212 4530 1.8 70.55 7.8 58.1 15 113417  
## 4 2110 3378 1.9 70.66 10.1 39.9 65 51945  
## 5 21198 5114 1.1 71.71 10.3 62.6 20 156361  
## 6 2541 4884 0.7 72.06 6.8 63.9 166 103766  
## Longitude Latitude Region  
## 1 -86.7509 32.5901 South  
## 2 -127.2500 49.2500 West  
## 3 -111.6250 34.2192 West  
## 4 -92.2992 34.7336 South  
## 5 -119.7730 36.5341 West  
## 6 -105.5130 38.6777 West

dim(state) # Number of observations and variables

## [1] 50 11

names(state) # Names of variables

## [1] "Population" "Income" "Illiteracy" "LifeExp"   
## [5] "Murder" "HighSchoolGrad" "Frost" "Area"   
## [9] "Longitude" "Latitude" "Region"

## Train-test Split

library(caTools)  
set.seed(12)  
# Randomly split the dataset with 70% in the training set  
spl <- sample.split(state$LifeExp, SplitRatio = 0.7)  
train <- state |> subset(spl == TRUE)  
test <- state |> subset(spl == FALSE)

## 7-variable Linear Regression Model

lm1 <- lm(LifeExp ~ Population + Murder + Frost + Income + Illiteracy + Area + HighSchoolGrad, data = train)  
  
lm1\_pred <- predict(lm1, newdata = test)  
# Out-of-sample R^2  
SSE <- sum((test$LifeExp - lm1\_pred) ^ 2)  
SST <- sum((test$LifeExp - mean(train$LifeExp)) ^ 2)  
R2\_lm1 <- 1 - SSE/SST  
R2\_lm1

## [1] 0.05283534

## 4-variable Linear Regression Model

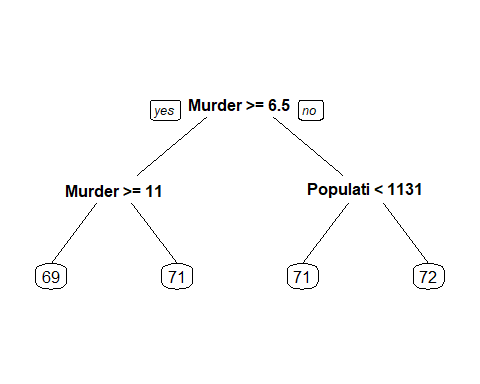
lm2 <- lm(LifeExp ~ Population + Murder + Frost + HighSchoolGrad, data = train)  
  
lm2\_pred <- predict(lm2, newdata = test)  
# Out-of-sample R^2  
SSE <- sum((test$LifeExp - lm2\_pred) ^ 2)  
SST <- sum((test$LifeExp - mean(train$LifeExp)) ^ 2)  
R2\_lm2 <- 1 - SSE/SST  
R2\_lm2

## [1] 0.6438655

By removing independent variables, the on the test test is increased, meaning the overfitting problem is alleviated. The equivalent procedure in CART is pruning to have a smaller tree.

## CART Model

library(rpart)  
library(rpart.plot)  
  
rtree <- rpart(LifeExp ~ Population + Murder + Frost + Income + Illiteracy + Area + HighSchoolGrad, data = train, method = "anova", minbucket = 5)  
  
prp(rtree) # Plot the tree



Independent variables Murder and Population appear in the tree. The CART model is easier to interpret.

## CART Prediction

rtree\_pred <- predict(rtree, newdata = test, type = "vector")  
# Out-of-sample R^2  
SSE <- sum((test$LifeExp - rtree\_pred) ^ 2)  
SST <- sum((test$LifeExp - mean(train$LifeExp)) ^ 2)  
R2\_rtree <- 1 - SSE/SST  
R2\_rtree

## [1] 0.1813543

## Random Forest

library(randomForest)

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

set.seed(1234)  
rf <- randomForest(LifeExp ~ Population + Murder + Frost + Income + Illiteracy + Area + HighSchoolGrad, data = train, ntree = 100, nodesize = 5)  
  
rf\_pred <- predict(rf, newdata = test)  
# Out-of-sample R^2  
SSE <- sum((test$LifeExp - rf\_pred) ^ 2)  
SST <- sum((test$LifeExp - mean(train$LifeExp)) ^ 2)  
R2\_rf <- 1 - SSE/SST  
R2\_rf

## [1] 0.6121284

## Best Model

# Out-of-sample R^2  
c("7-variable lm" = R2\_lm1, "4-variable lm" = R2\_lm2,  
 "Tree" = R2\_rtree, "Random Forest" = R2\_rf)

## 7-variable lm 4-variable lm Tree Random Forest   
## 0.05283534 0.64386555 0.18135431 0.61212838

The 4-variable linear regression model has the highest out-of-sample . The tree model is the easiest to interpret.

# Q2 Clustering

bow <- read.csv("DailyKos.csv")

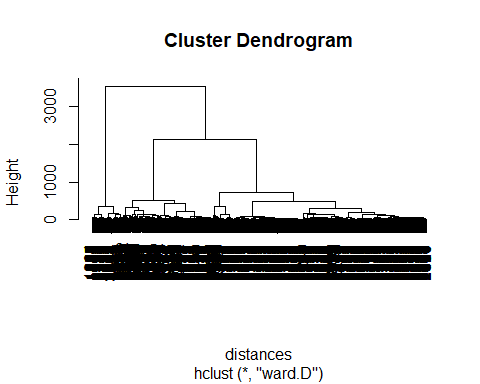
## Hierarchical Clustering

# Compute distances between points  
distances <- dist(bow, method = "euclidean")  
# Hierarchical clustering  
hbow <- hclust(distances, method = "ward.D")

Euclidean distance metrics is used to calculate distances.  
Hierarchical clustering takes lot of time because in each recursion, it calculates the distance of all combinations between every two data points.

### Plot the dendrogram

plot(hbow)



### Choose the Number of Clusters

10 clusters are recommended for different categories of articles.

no\_clusters <- 10  
# Cut the tree into 10 clusters  
h\_10clust <- cutree(hbow, no\_clusters)  
# No. of observations in each cluster  
table(h\_10clust)

## h\_10clust  
## 1 2 3 4 5 6 7 8 9 10   
## 1266 179 279 139 407 714 63 95 146 142

### Split the Clusters and Analyze Each Cluster

# Split the dataset into a dataset for each cluster  
# Find the six most frequent words in each cluster  
no\_clusters <- 10  
for (i in 1:no\_clusters){  
 bow |>  
 subset(h\_10clust == i) |> # Filter  
 colMeans() |> # Take the average of each column  
 sort(decreasing = TRUE) |>  
 head() |>   
 print.data.frame()  
 cat("\n") # Add a line for easier reading  
}

## [1] bush kerry democrat poll republican state   
## <0 rows> (or 0-length row.names)  
##   
## [1] november vote poll challenge bush republican  
## <0 rows> (or 0-length row.names)  
##   
## [1] democrat republican state bush parties senate   
## <0 rows> (or 0-length row.names)  
##   
## [1] kerry bush poll presided voter campaign  
## <0 rows> (or 0-length row.names)  
##   
## [1] bush iraq war administration presided   
## [6] american   
## <0 rows> (or 0-length row.names)  
##   
## [1] poll democrat elect kerry bush race   
## <0 rows> (or 0-length row.names)  
##   
## [1] dean kerry democrat campaign edward gephardt  
## <0 rows> (or 0-length row.names)  
##   
## [1] bush administration presided war iraq   
## [6] house   
## <0 rows> (or 0-length row.names)  
##   
## [1] kerry dean poll edward clark primaries  
## <0 rows> (or 0-length row.names)  
##   
## [1] november poll challenge democrat vote house   
## <0 rows> (or 0-length row.names)

There is a cluster that is mostly about the Iraq war. There are several clusters that are mostly about the democratic party.

## K-means Clustering

no\_clusters <- 10  
set.seed(23)  
kbow <- kmeans(bow, no\_clusters)  
k\_10clust <- kbow$cluster  
# No. of observations in each cluster  
table(k\_10clust)

## k\_10clust  
## 1 2 3 4 5 6 7 8 9 10   
## 46 280 43 142 293 195 1750 356 160 165

The number of observations in each cluster is different from hierarchical clustering, because the clustering algorithms are different.

### Split the Clusters and Analyze Each Cluster

# Split the dataset into a dataset for each cluster  
# Find the six most frequent words in each cluster  
no\_clusters <- 10  
for (i in 1:no\_clusters){  
 bow |>  
 subset(k\_10clust == i) |>  
 colMeans() |>  
 sort(decreasing = TRUE) |>  
 head() |>   
 print.data.frame()  
 cat("\n")  
}

## [1] democrat parties republican state seat senate   
## <0 rows> (or 0-length row.names)  
##   
## [1] bush administration presided time year   
## [6] house   
## <0 rows> (or 0-length row.names)  
##   
## [1] bush kerry presided iraq vote democrat  
## <0 rows> (or 0-length row.names)  
##   
## [1] dean kerry clark edward democrat primaries  
## <0 rows> (or 0-length row.names)  
##   
## [1] kerry bush poll campaign presided democrat  
## <0 rows> (or 0-length row.names)  
##   
## [1] iraq war bush iraqi american official  
## <0 rows> (or 0-length row.names)  
##   
## [1] bush poll kerry democrat general elect   
## <0 rows> (or 0-length row.names)  
##   
## [1] democrat republican state elect senate parties   
## <0 rows> (or 0-length row.names)  
##   
## [1] november poll challenge democrat vote house   
## <0 rows> (or 0-length row.names)  
##   
## [1] november vote poll challenge bush republican  
## <0 rows> (or 0-length row.names)

Overall, these two groups of clusters have very similar keywords, like “bush”, “kerry”, “republican”, “november”, “iraq”, etc.  
2 clusters starting with “november” among 10 clusters are identical with hierarchical clustering.