Project 06

Indrajith Wasala Mudiyanselage

Question 01

a)

The Variables actually used in tree construction are “CAtBat” “CHits” “AtBat” “CRuns” “Hits” “Walks” and “CRBI”. There are 9 nodes and residual mean deviance is 0.1694.The distribution of residuals is given below.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.7077 -0.2213  0.0353  0.0000  0.2303  1.7018
```

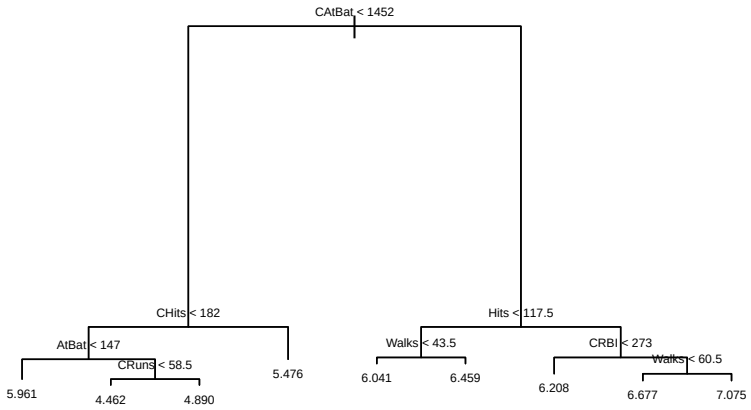

Figure 1: Regression tree for Hitters data

Let R_j be the partitions of the predictor space.

$$
\begin{aligned}
R_1 &= \{X \mid CAtBat < 1452, CHits < 182, AtBAt < 147\} \\
R_2 &= \{X \mid CAtBat < 1452, CHits < 182, AtBAt \geq 147, CRuns < 58.5\} \\
R_3 &= \{X \mid CAtBat < 1452, CHits < 182, AtBAt \geq 147, CRuns \geq 58.5\} \\
R_4 &= \{X \mid CAtBat < 1452, CHits \geq 182\} \\
R_5 &= \{X \mid CAtBat \geq 1452, Hits < 117.5, Walks < 43.5\} \\
R_6 &= \{X \mid CAtBat \geq 1452, Hits < 117.5, Walks \geq 43.5\} \\
R_7 &= \{X \mid CAtBat \geq 1452, Hits \geq 117.5, CRBI < 273\} \\
R_8 &= \{X \mid CAtBat \geq 1452, Hits \geq 117.5, CRBI \geq 273, Walks < 60.5\} \\
R_9 &= \{X \mid CAtBat \geq 1452, Hits \geq 117.5, CRBI \geq 273, Walks \geq 60.5\}
\end{aligned}
$$

The estimated test MSE using LOOCV is 0.2545162.

b)

The best-pruned tree and un-pruned tree happened to be the same. The number of terminal nodes is 9 in both cases. Therefore, the estimated test MSE for the best-pruned tree is the same as before (0.2545162). The most important variables are “CAtBat” “CHits” “AtBat” “CRuns” “Hits” “Walks” and “CRBI”.

Figure 2: Plot the estimated test error rate

c)

Since we are using a large B value (1000), the Out-of-Bag (OOB) error is appropriately equal to the LOOCV test error. Therefore test MSE is 0.1878328. According to the Node purity plot, we can see that "CAtBat" (IncNodePurity=78.9240659), "CRuns" (IncNodePurity=33.1447878) and "CHits"(IncNodePurity=27.0610915) are the most important predictors.

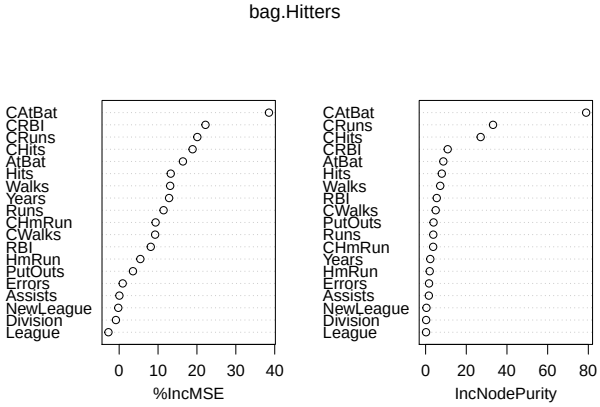

Figure 3: Variable importance measure for each predictor (Bagging)

d)

Since we are using a large B value (1000), the Out-of-Bag (OOB) error is appropriately equal to the LOOCV test error. Therefore test MSE is 0.1802431. According to the Node purity plot, we can see that "CAtBat" (IncNodePurity=40.2152018),"CHits"(IncNodePurity=35.2244118),"CRuns" (IncNodePurity=32.2252499) "CWalks" (IncNodePurity=19.5160848) and "CRBI" (IncNodePurity=18.3389603) are the most important predictors.

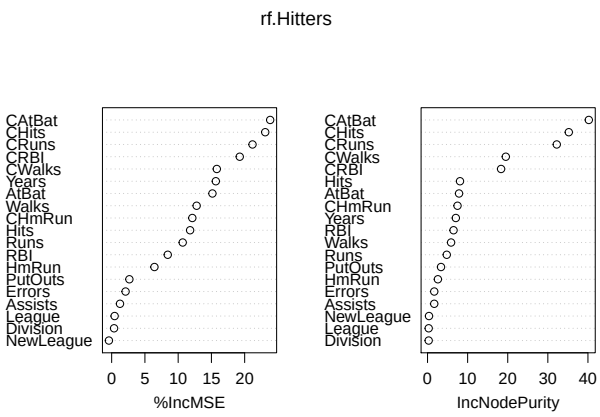

Figure 4: Variable importance measure for each predictor (RF)

e)

The estimated test MSE using boosting approach is 0.1540222. According to the relative influence, the most important predictors are "CAtBat" (Rel.inf=23.4492712),"CHits"(Rel.inf=14.3875368),"CRuns" (Rel.inf=12.2942197) and "CRBI" (Rel.inf=10.0409460).

f)

When comparing the MSE values, the boosting approach has the lowest MSE value (0.1540222). Therefore I recommend boosting approach to analyze this data with B=1000, d=1, and $\lambda = 0.01$. In my previous project, my recommendation (for this dataset) was the Ridge regression model as it has the lowest test MSE (0.3607). But, the test MSE of the boosting approach is still lesser than the Ridge regression model test MSE. Therefore boosting approach is better.

Question 02

a)

The optimal cost parameter for the support vector classifier is 4.6. There are 1051 support vectors. 524 support vectors correspond to region 0 (level 0) and, the remaining 527 correspond to region 1 (level 1). If we use 10-fold cross-validation the total accuracy is 77.45% (Single Accuracies: 80.5, 77, 81, 81, 75, 76, 77, 79.5, 75.5, 72). Therefore test error rate is 0.2255 (22.55%).

b)

The optimal cost parameter for the support vector machine with degree 2 polynomial is 4.9. There are 1217 support vectors. 600 support vectors correspond to region 0 (level 0) and, the remaining 617 correspond to region 1 (level 1). If we use 10-fold cross-validation the total accuracy is 73.1 % (Single Accuracies: 73.5, 72, 71.5, 74.5, 74.5, 71.5, 74.5, 74, 71, 74). Therefore test error rate is 0.269 (26.9%).

c)

The optimal cost parameter for the support vector machine with a radial kernel is 0.8 (Optimal gamma is 4.5). There are 1024 support vectors. 507 support vectors correspond to region 0 (level 0) and, the remaining 517 correspond to region 1 (level 1). If we use 10-fold cross-validation the total accuracy is 98.5 % (Single Accuracies: 100, 98, 98.5, 98, 98, 98, 97, 99, 99.5, 99). Therefore test error rate is 0.015 (1.5%).

d)

When comparing the test error rates, the support vector machine with a radial kernel has the lowest test error rates (0.015). Therefore I recommend the support vector machine with a radial kernel for these data. But, note that this may course an overfitting problem. Therefore, further investigation into the model is needed. By looking at projects 3 and 4 recommendations, my recommendation was the KNN model with K=6. The test error for KNN was 0.1615. This value is greater than the test error rate of the support vector machine with a radial kernel. Therefore, support vector machine with a radial kernel is still better.

R Codes

```
## ----include=FALSE-----------------------------------------------
#########################################################################################################################
# Question 01

library(ISLR)
Hitters.new<-na.omit(Hitters)
Hitters.new$Salary<- log(Hitters.new$Salary)
str(Hitters.new)


## ----include=FALSE-----------------------------------------------
## a)

library(tree)

tree.Hitters <- tree(Salary ~ ., Hitters.new)
sumry<-summary(tree.Hitters)



## ----echo=FALSE--------------------------------------------------
summary(sumry$residuals)


## ----echo=FALSE,fig.align="center",fig.cap="Regression tree for Hitters data",  out.width = "50%"----
# Plot the tree
plot(tree.Hitters)
text(tree.Hitters, pretty = 0, cex = 0.5)


## ----include=FALSE-----------------------------------------------
LOOCV<-function(dataset){
n<-length(dataset[,1])
lr.pred.fit<-c()
for (i in 1:n) {
  newdata<-dataset[-i,]
  testdata<-dataset[i,]
  fit <- tree(Salary ~ ., newdata)
  lr.pred.fit[i] <- predict(fit, testdata)
}
  MSE<- mean((lr.pred.fit - dataset$Salary)^2)
  return(list(MSE=MSE))
}
test.MSE<-LOOCV(dataset=Hitters.new)
test.MSE


## ----include=FALSE-----------------------------------------------
## b)

set.seed(6340)
cv.Hitters <- cv.tree(tree.Hitters, FUN = prune.tree, K=263)
best.pruned <- which.min(cv.Hitters$size)


## ----echo=FALSE,fig.align="center",fig.cap="Plot the estimated test error rate",  out.width = "40%"----
plot(cv.Hitters$size, cv.Hitters$dev, type = "b")


## ----include=FALSE-----------------------------------------------
## c)
```

```r
library(randomForest)

set.seed(1)
bag.Hitters <- randomForest(Salary ~ ., data = Hitters.new, mtry = 19, ntree = 1000, importance = TRUE)

# Out-of-bag Mean of squared residuals
yhat.bag <- predict(bag.Hitters)
mean((yhat.bag - Hitters.new$Salary)^2)
importance(bag.Hitters)

## ----echo=FALSE,fig.align="center",fig.cap="Variable importance measure for each predictor (Bagging)",  out.wi
varImpPlot(bag.Hitters)


## ----include=FALSE-----------------------------------------
## d)

set.seed(1)
rf.Hitters <- randomForest(Salary ~ ., data = Hitters.new, mtry = 19/3, ntree = 1000, importance = TRUE)

# Out-of-bag Mean of squared residuals
yhat.rf <- predict(rf.Hitters)
mean((yhat.rf - Hitters.new$Salary)^2)
importance(rf.Hitters)


## ----echo=FALSE,fig.align="center",fig.cap="Variable importance measure for each predictor (RF)",  out.width =
varImpPlot(rf.Hitters)


## ---- cache=TRUE,include=FALSE-----------------------------
## e)

library(gbm)
set.seed(1)
boost.Hitters <- gbm(Salary ~ ., data = Hitters.new, distribution = "gaussian",
    n.trees = 1000, interaction.depth = 1,shrinkage=0.01,cv.folds=263)


## ----include=FALSE-----------------------------------------
library(gbm)
yhat.boost <- predict(boost.Hitters, newdata = Hitters.new, n.trees = 1000)
mean((yhat.boost - Hitters.new$Salary)^2)


## ----include=FALSE-----------------------------------------
summary(boost.Hitters)


## ----include=FALSE-----------------------------------------
##########################################################################################################
# Question 02

diabetes<-read.csv("diabetes.csv",header = T)
diabetes$Outcome<-as.factor(diabetes$Outcome)
str(diabetes)


## ---- include=FALSE---------------------------------------
## a)

library(e1071)
set.seed(1)
```

```
tune.out <- tune(svm, Outcome ~ ., data = diabetes, kernel = "linear", ranges = list(cost = c(4,4.1,4.2,4.3,4.4,
summary(tune.out)


## ----include=FALSE-----------------------------------------
bestmod <- tune.out$best.model
summary(bestmod)


## ----include=FALSE-----------------------------------------
# Compute test error using 10- fold CV
set.seed(1)
svmfit <- svm(Outcome ~ ., data = diabetes, kernel = "linear", cost = 4.6, scale = TRUE, cross=10)
summary(svmfit)


## ---- include=FALSE---------------------------------------
## b)

library(e1071)
set.seed(1)
b.tune.out <- tune(svm, Outcome ~ ., data = diabetes, kernel = "polynomial",degree=2, ranges = list(cost = c(4.6
summary(b.tune.out)


## ----include=FALSE-----------------------------------------
bestmod.b <- b.tune.out$best.model
summary(bestmod.b)


## ----include=FALSE-----------------------------------------
# Compute test error using 10- fold CV
set.seed(1)
svmfit.b <- svm(Outcome ~ ., data = diabetes, kernel = "polynomial",degree=2, cost = 4.9, scale = TRUE, cross=10
summary(svmfit.b)


## ---- cache=TRUE, include=FALSE---------------------------
## b)

library(e1071)
set.seed(1)
c.tune.out <- tune(svm, Outcome ~ ., data = diabetes, kernel = "radial", ranges = list(cost = c(0.5,0.6,0.7,0.8,
summary(c.tune.out)


## ----include=FALSE-----------------------------------------
bestmod.c <- c.tune.out$best.model
summary(bestmod.c)


## ----include=FALSE-----------------------------------------
# Compute test error using 10- fold CV
set.seed(1)
svmfit.c <- svm(Outcome ~ ., data = diabetes, kernel = "radial", gamma = 4.5, cost = 0.8, scale = TRUE, cross=10
summary(svmfit.c)
```