Data Mining and Predictive Modelling
Assignment 6

Name: Rushikesh Jyoti
Division: A
Roll no: 27
SRN: 201901139

Question

- Read the dataset **"prostrate.csv"** that is provided to you.
- Build a suitable decision tree predictive model to predict **Tumor Log Volume** based on predictor information.
- Plot the decision tree and develop some metrics to determine the accuracy of your model. (Compute various evaluation parameters of the tree model built).
- Cross validate and optimize the model using hold back V-fold technique.

Use method of pruning to avoid over-fitting and derive the best size of the decision tree.

Code

```
setwd("C:\\VS_Workshop\\Sem 6\\Data Mining and Predictive Modelling\\Assignments\\Ass6") # nolint

library(datasets)
library(caTools)
library(caret)
library(Metrics)
library(party)
library(dplyr)
library(magrittr)
library(tree)
library(corrplot)

df = read.csv("./prostate.csv")
head(df)
```

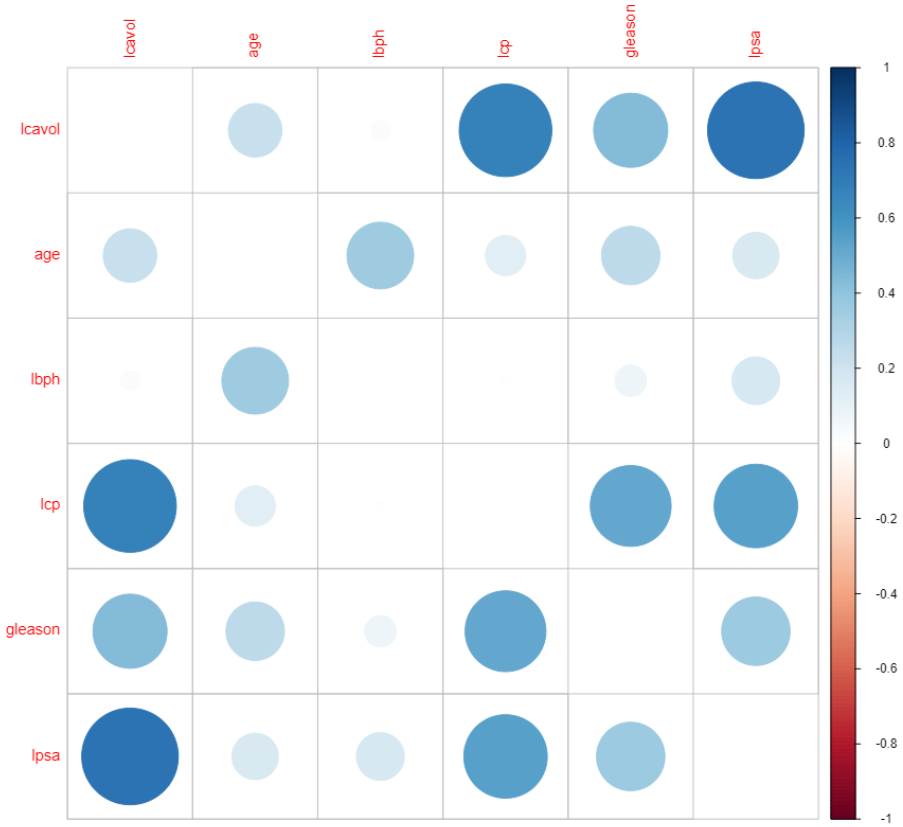
```r
corrplot(cor(df), diag=FALSE)

# lbph doesnt have enough correlation with lcavol so drop it
df = select(df, -lbph)
head(df)

#Splitting dataset into 4:1 or 80:20 ratio for train and test data
sample_data <- sample.split(df, SplitRatio = 0.8)
train_data <- subset(df, sample_data == TRUE)
test_data <- subset(df, sample_data == FALSE)

# Create the decision tree model using ctree and plot the model
model <- tree(lcavol ~ ., train_data, mincut=1)
# The minimum number of observations
# to include in either child node = 1
model
plot(model)

# Pruning the tree
prune.tree(model)

cut_model = prune.tree(model, k=9)
cut_model
plot(cut_model)

predictions = predict(model, test_data)
# predictions = predict(cut_model, test_data)
predictions

print(test_data$lcavol)
print(predictions)

errors = function(pred) {
  mae <- MAE(test_data$lcavol, pred)
  mse <- mse(test_data$lcavol, pred)
  rmse <- RMSE(test_data$lcavol, pred)
  r2 <- R2(test_data$lcavol, pred)

  cat("\nMean Absolute Error:", mae, "\nMean Squared Error:", mse, "\nRMSE:", rmse, "\nR-squared:",
r2, "\n")
}

errors(predict(model, test_data))
errors(predict(cut_model, test_data))
```

Output

Taking a look at our dataset

```
> df = read.csv("./prostate.csv")

> head(df)
      lcavol age      lbph       lcp gleason       lpsa
1 -0.5798185  50 -1.386294 -1.386294       6 -0.4307829
2 -0.9942523  58 -1.386294 -1.386294       6 -0.1625189
3 -0.5108256  74 -1.386294 -1.386294       7 -0.1625189
4 -1.2039728  58 -1.386294 -1.386294       6 -0.1625189
5  0.7514161  62 -1.386294 -1.386294       6  0.3715636
6 -1.0498221  50 -1.386294 -1.386294       6  0.7654678
```

Correlation Plot of dataset



Since lbph is not correlated with lcavol we shall remove it

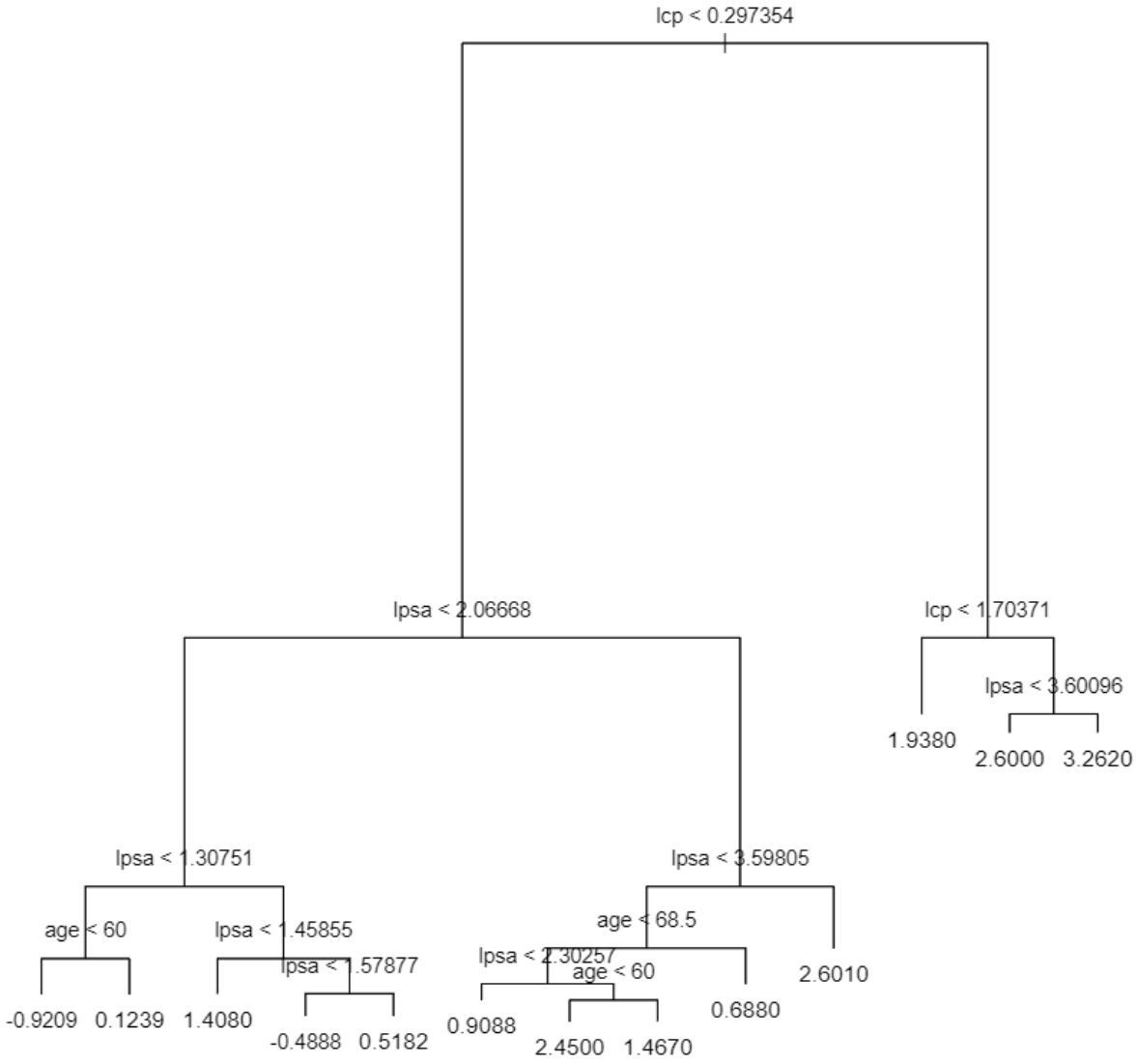
Split the data into 80:20 ratio

```
> #Splitting dataset into 4:1 or 80:20 ratio for train and test data
> sample_data <- sample.split(df, SplitRatio = 0.8)
> train_data <- subset(df, sample_data == TRUE)
> test_data <- subset(df, sample_data == FALSE)
> dim(train_data)
[1] 78  5
> dim(test_data)
[1] 19  5
```

Creating the model

```
> # Create the decision tree model using ctree and plot the model
> model <- tree(lcavol ~ ., train_data, mincut=1)
> # The minimum number of observations
> # to include in either child node = 1
> model
node), split, n, deviance, yval
      * denotes terminal node

 1) root 78 110.80000  1.3780
   2) lcp < 0.297354 49  52.60000  0.7893
     4) lpsa < 2.06668 25  18.28000  0.1778
       8) lpsa < 1.30751 10   5.92900 -0.3985
        16) age < 60 5   0.23930 -0.9209 *
        17) age > 60 5   2.96100  0.1239 *
       9) lpsa > 1.30751 15   6.81500  0.5619
        18) lpsa < 1.45855 3   0.06331  1.4080 *
        19) lpsa > 1.45855 12   4.06700  0.3504
          38) lpsa < 1.57877 2   0.01075 -0.4888 *
          39) lpsa > 1.57877 10   2.36600  0.5182 *
     5) lpsa > 2.06668 24  15.24000  1.4260
      10) lpsa < 3.59805 21   9.75200  1.2580
        20) age < 68.5 15   4.89600  1.4870
          40) lpsa < 2.30257 3   0.25510  0.9088 *
          41) lpsa > 2.30257 12   3.39000  1.6310
            82) age < 60 2   0.99110  2.4500 *
            83) age > 60 10   0.78990  1.4670 *
        21) age > 68.5 6   2.12400  0.6880 *
      11) lpsa > 3.59805 3   0.75400  2.6010 *
   3) lcp > 0.297354 29  12.55000  2.3720
     6) lcp < 1.70371 15   3.42900  1.9380 *
     7) lcp > 1.70371 14   3.27000  2.8370
      14) lpsa < 3.60096 9   1.17100  2.6000 *
      15) lpsa > 3.60096 5   0.69040  3.2620 *
```

Plotting our model



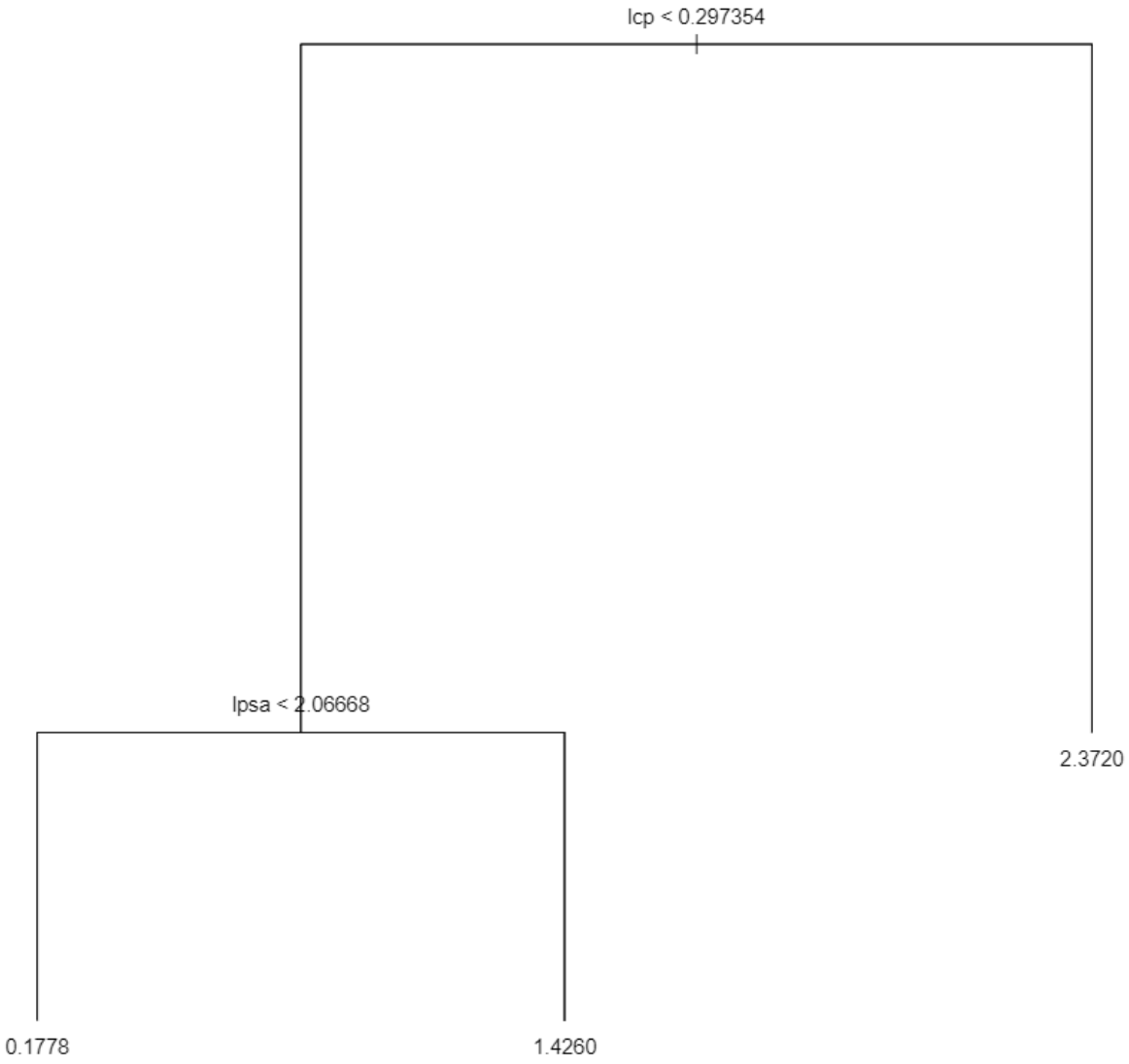
Let’s check some pruning status of our tree model

```
> prune.tree(model)
$size
 [1] 13 12 10  9  8  7  6  5  4  3  2  1

$dev
 [1]  15.84339  17.25237  20.11261  21.80288  24.48764  27.21633  29.94883
 [8]  34.68205  40.21672  46.06304  65.14934 110.77488

$k
 [1]      -Inf  1.408977  1.430119  1.690276  2.684757  2.728690  2.732500
 [8]  4.733223  5.534673  5.846318 19.086301 45.625539

$method
[1] "deviance"

attr(,"class")
[1] "prune"         "tree.sequence"
```

Let’s try pruning the tree with k=9

```
> cut_model = prune.tree(model, k=9)
> cut_model
node), split, n, deviance, yval
      * denotes terminal node

1) root 78 110.80 1.3780
  2) lcp < 0.297354 49  52.60 0.7893
    4) lpsa < 2.06668 25  18.28 0.1778 *
    5) lpsa > 2.06668 24  15.24 1.4260 *
  3) lcp > 0.297354 29  12.55 2.3720 *
```

And we plot the pruned tree



Let’s use this custom function for errors (MAE, MSE, RMSE, R^2)

```
> errors = function(pred) {
+     mae <- MAE(test_data$lcavol, pred)
+     mse <- mse(test_data$lcavol, pred)
+     rmse <- RMSE(test_data$lcavol, pred)
+     r2 <- R2(test_data$lcavol, pred)
+
+     cat("\nMean Absolute Error:", mae, "\nMean Squared Error:", mse, "\nRMSE$
+ }
```

Metrics of our original model

```
> errors(predict(model, test_data))

Mean Absolute Error: 0.5419817
Mean Squared Error: 0.4260876
RMSE: 0.6527539
R-squared: 0.7194934
```

Metrics of pruned tree

```
> errors(predict(cut_model, test_data))

Mean Absolute Error: 0.7091679
Mean Squared Error: 0.7567918
RMSE: 0.8699378
R-squared: 0.4836515
```