Homework 7 - Decision Tree

Shiloh Bradley

6/23/2020

Toyota Corolla

```
df <- read.csv("ToyotaCorolla2.csv")
head(df)
```

```
##   Price Age    KM Fuel_Type HP Met_Color Automatic   cc Doors
## 1 13500  23 46986    Diesel 90         1         0 2000     3
## 2 13750  23 72937    Diesel 90         1         0 2000     3
## 3 13950  24 41711    Diesel 90         1         0 2000     3
## 4 14950  26 48000    Diesel 90         0         0 2000     3
## 5 13750  30 38500    Diesel 90         0         0 2000     3
## 6 12950  32 61000    Diesel 90         0         0 2000     3
##   Quarterly_Tax Weight
## 1           210   1165
## 2           210   1165
## 3           210   1165
## 4           210   1165
## 5           210   1170
## 6           210   1170
```

```
summary(df)
```

```
##      Price            Age              KM           Fuel_Type
##  Min.   : 4350   Min.   : 1.00   Min.   :     1   CNG   :  17
##  1st Qu.: 8450   1st Qu.:44.00   1st Qu.: 43000   Diesel: 155
##  Median : 9900   Median :61.00   Median : 63390   Petrol:1264
##  Mean   :10731   Mean   :55.95   Mean   : 68533
##  3rd Qu.:11950   3rd Qu.:70.00   3rd Qu.: 87021
##  Max.   :32500   Max.   :80.00   Max.   :243000
##        HP          Met_Color        Automatic             cc
##  Min.   : 69.0   Min.   :0.0000   Min.   :0.00000   Min.   : 1300
##  1st Qu.: 90.0   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.: 1400
##  Median :110.0   Median :1.0000   Median :0.00000   Median : 1600
##  Mean   :101.5   Mean   :0.6748   Mean   :0.05571   Mean   : 1577
##  3rd Qu.:110.0   3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.: 1600
##  Max.   :192.0   Max.   :1.0000   Max.   :1.00000   Max.   :16000
##      Doors       Quarterly_Tax        Weight
##  Min.   :2.000   Min.   : 19.00   Min.   :1000
##  1st Qu.:3.000   1st Qu.: 69.00   1st Qu.:1040
##  Median :4.000   Median : 85.00   Median :1070
##  Mean   :4.033   Mean   : 87.12   Mean   :1072
##  3rd Qu.:5.000   3rd Qu.: 85.00   3rd Qu.:1085
##  Max.   :5.000   Max.   :283.00   Max.   :1615
```

```
M <- .25 * nrow(df)
#to be able to replicate the results, set initial seed for random
#number generator
set.seed(11317)
```

```r
holdout <- sample(1:nrow(df), M, replace = F)

df.train <- df[-holdout, ]   # Training set
df.test <- df[holdout, ]    # Test set
dim(df.train) #  1077 11
```

```
## [1] 1077   11
```

```r
dim(df.test)  #  359 11
```

```
## [1] 359  11
```

```r
lm1 <- lm(Price ~ .,
          data = df.train)
summary(lm1)
```

```
##
## Call:
## lm(formula = Price ~ ., data = df.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10455.9   -743.2    -39.2    683.9   6880.7
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -5.926e+03  1.370e+03  -4.326 1.66e-05 ***
## Age             -1.244e+02  2.993e+00 -41.560  < 2e-16 ***
## KM              -1.623e-02  1.489e-03 -10.901  < 2e-16 ***
## Fuel_TypeDiesel  1.104e+03  4.394e+02   2.514   0.0121 *
## Fuel_TypePetrol  2.857e+03  4.352e+02   6.563 8.22e-11 ***
## HP               2.283e+01  4.021e+00   5.679 1.74e-08 ***
## Met_Color        4.373e+01  8.695e+01   0.503   0.6151
## Automatic        3.792e+02  1.789e+02   2.119   0.0343 *
## cc              -2.157e-02  9.109e-02  -0.237   0.8129
## Doors           -2.619e+01  4.534e+01  -0.578   0.5636
## Quarterly_Tax    1.199e+01  1.850e+00   6.485 1.36e-10 ***
## Weight           1.753e+01  1.327e+00  13.207  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1314 on 1065 degrees of freedom
## Multiple R-squared:  0.8634, Adjusted R-squared:  0.862
## F-statistic: 612.1 on 11 and 1065 DF,  p-value: < 2.2e-16
```

```r
vif(lm1)
```

```
##                   GVIF Df GVIF^(1/(2*Df))
## Age           1.875729  1        1.369573
## KM            2.041341  1        1.428755
## Fuel_Type     6.903487  2        1.620941
## HP            2.177556  1        1.475654
## Met_Color     1.020219  1        1.010059
## Automatic     1.083180  1        1.040759
## cc            1.181299  1        1.086876
## Doors         1.168824  1        1.081122
```

```
## Quarterly_Tax 3.492318  1        1.868774
## Weight        3.017700  1        1.737153
```

There aren't any variables with VIF greater than 5, so we don't need to worry about those.

We will just worry about dropping insignificant values instead.

```r
lm2 <- lm(Price ~ . -Met_Color -cc -Doors, data = df.train)
summary(lm2)
```

```
##
## Call:
## lm(formula = Price ~ . - Met_Color - cc - Doors, data = df.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10329.8   -747.5    -57.7    686.9   6923.8
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -5.756e+03  1.343e+03  -4.286 1.99e-05 ***
## Age              -1.244e+02  2.988e+00 -41.637  < 2e-16 ***
## KM               -1.632e-02  1.483e-03 -11.008  < 2e-16 ***
## Fuel_TypeDiesel   1.108e+03  4.334e+02   2.558   0.0107 *
## Fuel_TypePetrol   2.837e+03  4.339e+02   6.538 9.65e-11 ***
## HP                2.291e+01  3.927e+00   5.835 7.13e-09 ***
## Automatic         3.857e+02  1.763e+02   2.187   0.0289 *
## Quarterly_Tax     1.191e+01  1.843e+00   6.465 1.54e-10 ***
## Weight            1.729e+01  1.259e+00  13.728  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1313 on 1068 degrees of freedom
## Multiple R-squared:  0.8634, Adjusted R-squared:  0.8623
## F-statistic: 843.5 on 8 and 1068 DF,  p-value: < 2.2e-16
```

```r
vif(lm2)
```

```
##                   GVIF Df GVIF^(1/(2*Df))
## Age           1.874295  1        1.369049
## KM            2.028835  1        1.424372
## Fuel_Type     6.395945  2        1.590289
## HP            2.081820  1        1.442851
## Automatic     1.054496  1        1.026887
## Quarterly_Tax 3.474634  1        1.864037
## Weight        2.723620  1        1.650340
```

```r
lmF <- lm2
```

```r
pred.lmF_test <- predict(lmF, df.test)
reslmF_test <- df.test$Price - pred.lmF_test
```

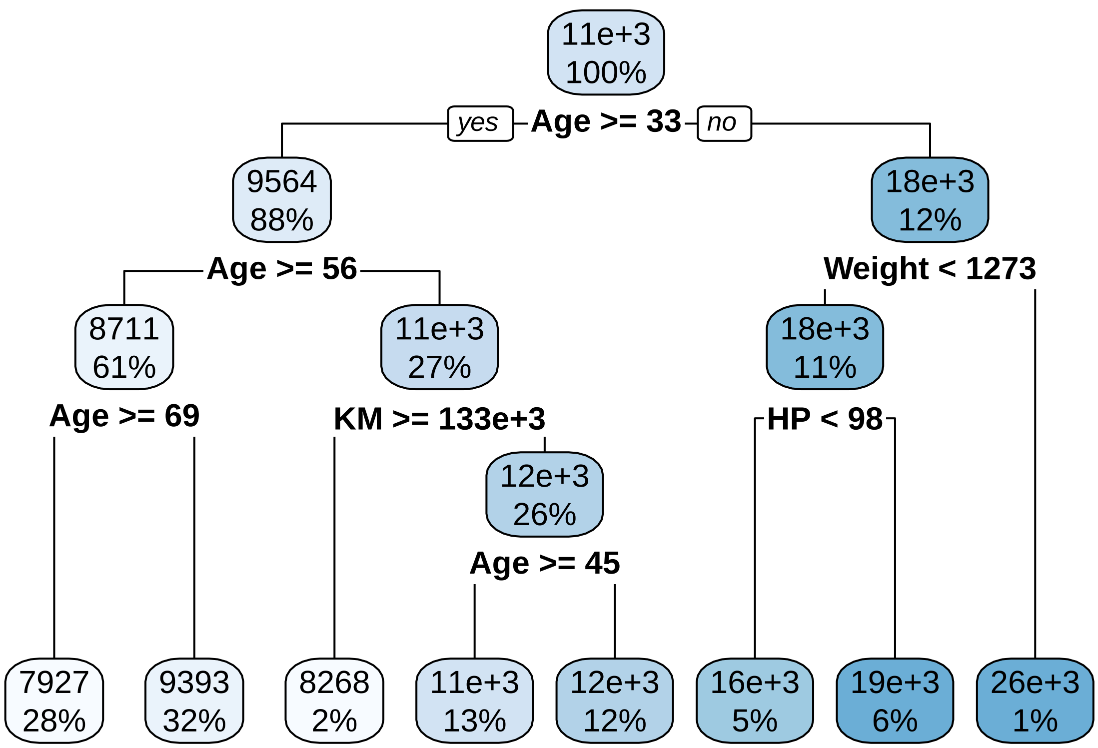
```r
R_train <- cor(df.train$Price, lmF$fitted.values)
R_train2 <- R_train**2
MSE.lmF_train <- sum(lmF$residuals**2)/nrow(df.train)
```

```r
RMSE.lmF_train <- sqrt(MSE.lmF_train)

R_test <- cor(df.test$Price, pred.lmF_test)
R_test2 <- R_test**2
MSE.lmF_test <- sum(reslmF_test**2)/nrow(df.test)
RMSE.lmF_test <- sqrt(MSE.lmF_test)
```

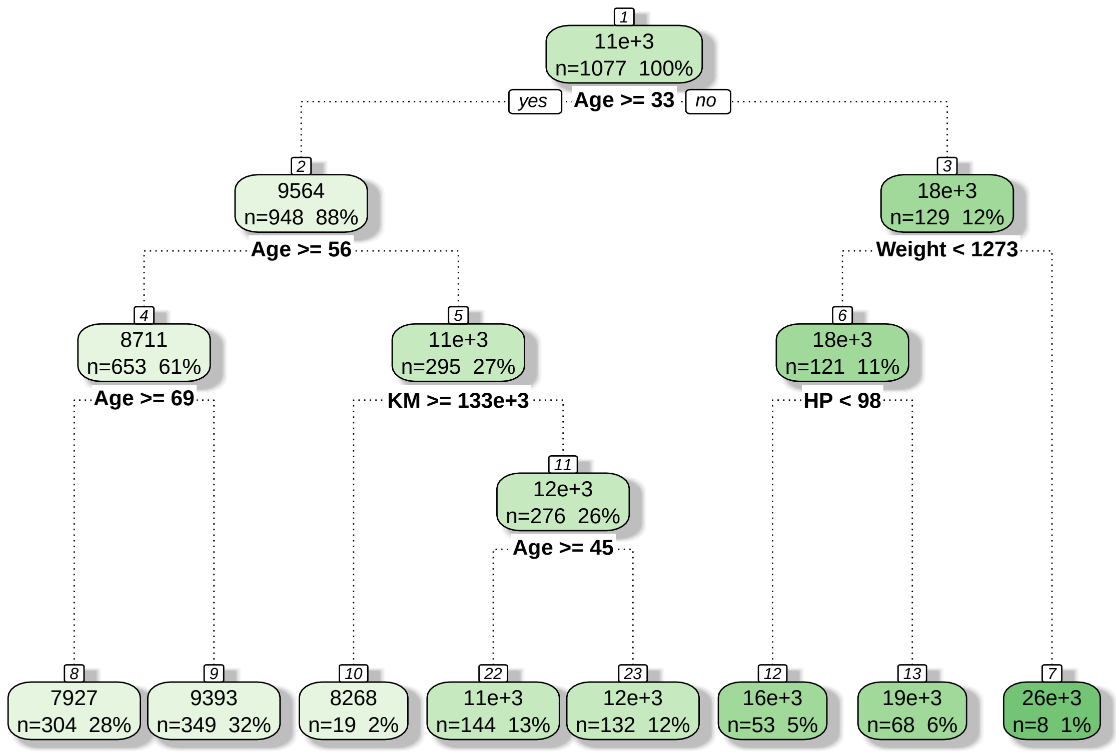
```r
RMSE <- c(RMSE.lmF_train, RMSE.lmF_test)
R2 <- c(R_train2, R_test2)
df.lmF <- rbind.data.frame(RMSE, R2)
colnames(df.lmF) <- c("training", "test")
rownames(df.lmF) <- c("RMSE.LM", "R_Square.LM")
df.lmF
```

```
##                  training         test
## RMSE.LM      1307.2017712 1346.4283848
## R_Square.LM     0.8633596    0.8805802
```

```r
fit <- rpart(Price ~ . -Met_Color -cc -Doors, data = df.train, method = "anova")
rpart.plot(fit)
```




```r
fancyRpartPlot(fit)
```



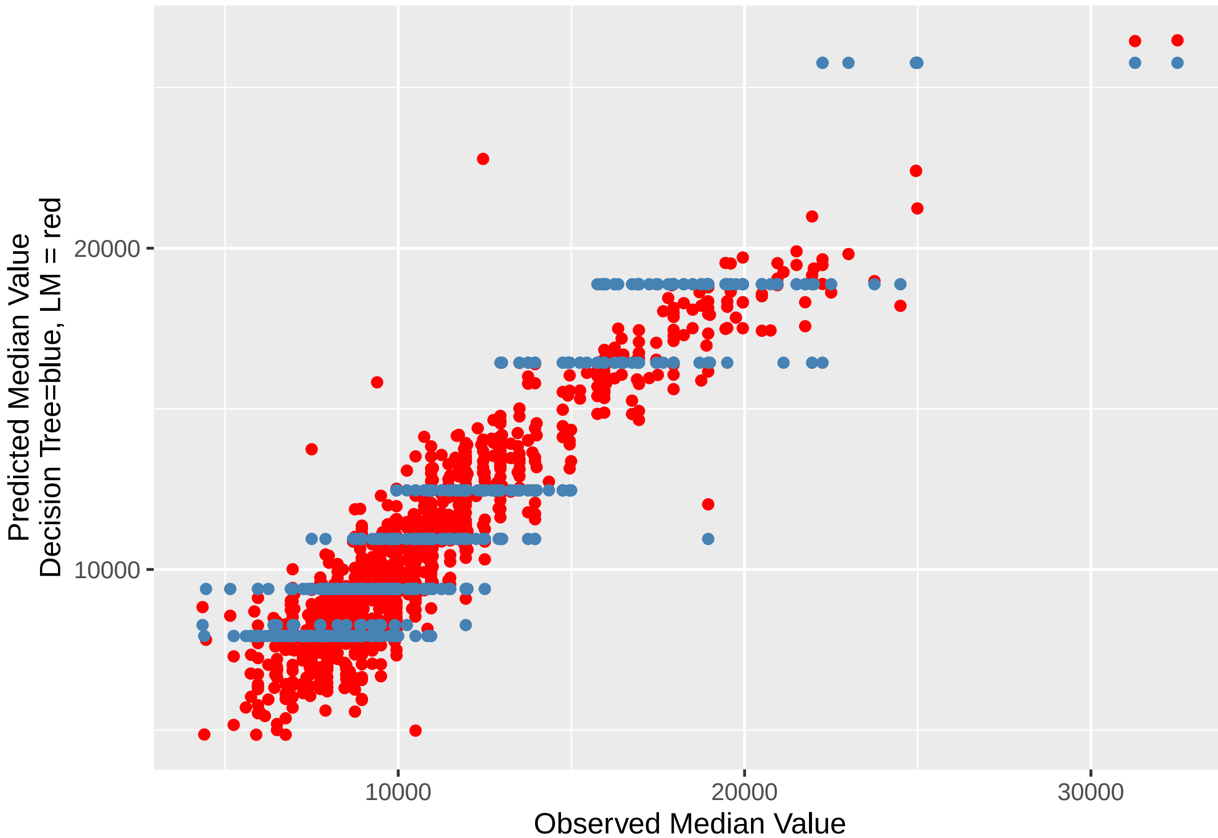
Rattle 2020−Jun−24 19:18:11 Shiloh

```r
pred.train_dt <- predict(fit, newdata = df.train)
pred.test_dt <- predict(fit, newdata = df.test)
MSE.train_dt <- sum((pred.train_dt - df.train$Price)**2)/nrow(df.train)
RMSE.train_dt <- sqrt(MSE.train_dt)
MSE.test_dt <- sum((pred.test_dt - df.test$Price)**2)/nrow(df.test)
RMSE.test_dt <- sqrt(MSE.test_dt)

r.train_dt <- cor(df.train$Price, pred.train_dt)

RMSE_dt <- c(RMSE.train_dt, RMSE.test_dt)

r.test_dt <- cor(df.test$Price, pred.test_dt)
r_dt <- c(r.train_dt, r.test_dt)
r2_dt <- r_dt**2

df.dt <- rbind.data.frame(RMSE_dt, r2_dt)
colnames(df.dt) <- c("training","test")
rownames(df.dt) <- c("RMSE.DT","R_Square.DT")
round(df.dt,2)
```

```
##              training    test
## RMSE.DT       1329.75 1443.30
## R_Square.DT      0.86    0.86
```

```r
df.train <- cbind.data.frame(df.train$Price, lmF$fitted.values, pred.train_dt)
colnames(df.train) <- c("Price_obs", "Price_lm.pred", "Price_dt.pred")
df.test <- cbind.data.frame(df.test$Price, pred.lmF_test, pred.test_dt)
colnames(df.test) <- c("Price_obs", "Price_lm.pred", "Price_dt.pred")
```

```
p.train1 <- ggplot(df.train, aes(x = Price_obs)) +
  geom_point(aes(y = Price_lm.pred), color = "red") +
  geom_point(aes(y = Price_dt.pred), color = "steelblue") +
  xlab("Observed Median Value") +
  ylab("Predicted Median Value\nDecision Tree=blue, LM = red")

p.train1
```



Charles Book Club

```
df <- read.csv("Charles_BookClub.csv", header = TRUE)
```

```
M <- .25 * nrow(df)
#to be able to replicate the results, set initial seed for random
#number generator
set.seed(11317)
holdout <- sample(1:nrow(df), M, replace = F)

df.train <- df[-holdout, ]    # Training set
df.test <- df[holdout, ]     # Test set
dim(df.train) #  1500 18
```

```
## [1] 1500   18
```

```
dim(df.test)  #  500 18
```

```
## [1] 500  18
```

```r
lm1 <- lm(Florence ~ .,
          data = df.train)
# summary(lm1)
vif(lm1)
```

```
##         Seq.          ID.      Gender            M            R            F
## 3499.030319 3498.938281    1.005187     1.382630     3.365684    25.170685
##  FirstPurch     ChildBks    YouthBks      CookBks     DoltYBks       RefBks
##   10.205032     3.404586    1.964167     3.628549     2.237993     2.068220
##      ArtBks      GeogBks    ItalCook   ItalHAtlas      ItalArt
##    2.108610     2.623941    1.666955     1.487643     1.648895
```

Remove variables with VIF greater than 5.

```r
lm2 <- lm(Florence ~ . -Seq. -ID. -FirstPurch, data = df.train)
summary(lm2)
```

```
##
## Call:
## lm(formula = Florence ~ . - Seq. - ID. - FirstPurch, data = df.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62351 -0.13810 -0.07681  0.00542  1.04590
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.620e-01  3.057e-02   5.299 1.34e-07 ***
## Gender      -8.415e-02  1.658e-02  -5.075 4.36e-07 ***
## M            4.476e-06  8.682e-05   0.052 0.958887
## R           -3.673e-03  1.243e-03  -2.955 0.003172 **
## F            3.597e-02  1.016e-02   3.539 0.000414 ***
## ChildBks    -5.388e-02  1.371e-02  -3.929 8.91e-05 ***
## YouthBks    -5.287e-02  1.772e-02  -2.983 0.002899 **
## CookBks     -6.264e-02  1.330e-02  -4.712 2.69e-06 ***
## DoltYBks    -6.587e-02  1.532e-02  -4.300 1.82e-05 ***
## RefBks      -8.097e-03  1.851e-02  -0.438 0.661795
## ArtBks       9.221e-02  1.782e-02   5.174 2.60e-07 ***
## GeogBks      1.894e-02  1.518e-02   1.248 0.212399
## ItalCook     9.575e-03  2.500e-02   0.383 0.701795
## ItalHAtlas   1.719e-03  4.530e-02   0.038 0.969743
## ItalArt      4.078e-02  4.200e-02   0.971 0.331718
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2895 on 1485 degrees of freedom
## Multiple R-squared:  0.1291, Adjusted R-squared:  0.1209
## F-statistic: 15.73 on 14 and 1485 DF,  p-value: < 2.2e-16
```

```r
vif(lm2)
```

```
##     Gender          M          R          F   ChildBks   YouthBks
##   1.004759   1.378995   1.890931  22.052163   3.340340   1.935618
##    CookBks   DoltYBks     RefBks     ArtBks    GeogBks   ItalCook
##   3.543613   2.208253   2.008012   2.072295   2.548580   1.659966
```

```
## ItalHAtlas    ItalArt
##   1.466824   1.644772
```

Remove insignificant variables.

```
lmF <- lm(Florence ~ . -Seq. -ID. -FirstPurch -M -RefBks -GeogBks -ItalCook -ItalHAtlas -ItalArt, data =
summary(lmF)
```

```
##
## Call:
## lm(formula = Florence ~ . - Seq. - ID. - FirstPurch - M - RefBks -
##     GeogBks - ItalCook - ItalHAtlas - ItalArt, data = df.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62292 -0.14093 -0.07659  0.00674  1.04768
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.151279   0.023282   6.498 1.11e-10 ***
## Gender      -0.085055   0.016561  -5.136 3.18e-07 ***
## R           -0.003140   0.001048  -2.995 0.002789 **
## F            0.041766   0.006341   6.587 6.21e-11 ***
## ChildBks    -0.057895   0.011969  -4.837 1.45e-06 ***
## YouthBks    -0.058811   0.016050  -3.664 0.000257 ***
## CookBks     -0.065665   0.011190  -5.868 5.42e-09 ***
## DoltYBks    -0.070769   0.013594  -5.206 2.20e-07 ***
## ArtBks       0.092659   0.015508   5.975 2.87e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2894 on 1491 degrees of freedom
## Multiple R-squared:  0.1266, Adjusted R-squared:  0.1219
## F-statistic:    27 on 8 and 1491 DF,  p-value: < 2.2e-16
```

```
vif(lmF)
```

```
##   Gender        R        F ChildBks YouthBks  CookBks DoltYBks   ArtBks
## 1.003253 1.346521 8.591445 2.547548 1.589178 2.512888 1.741012 1.570854
```

```
pred.lmF_test <- predict(lmF, df.test)
reslmF_test <- df.test$Florence - pred.lmF_test
```

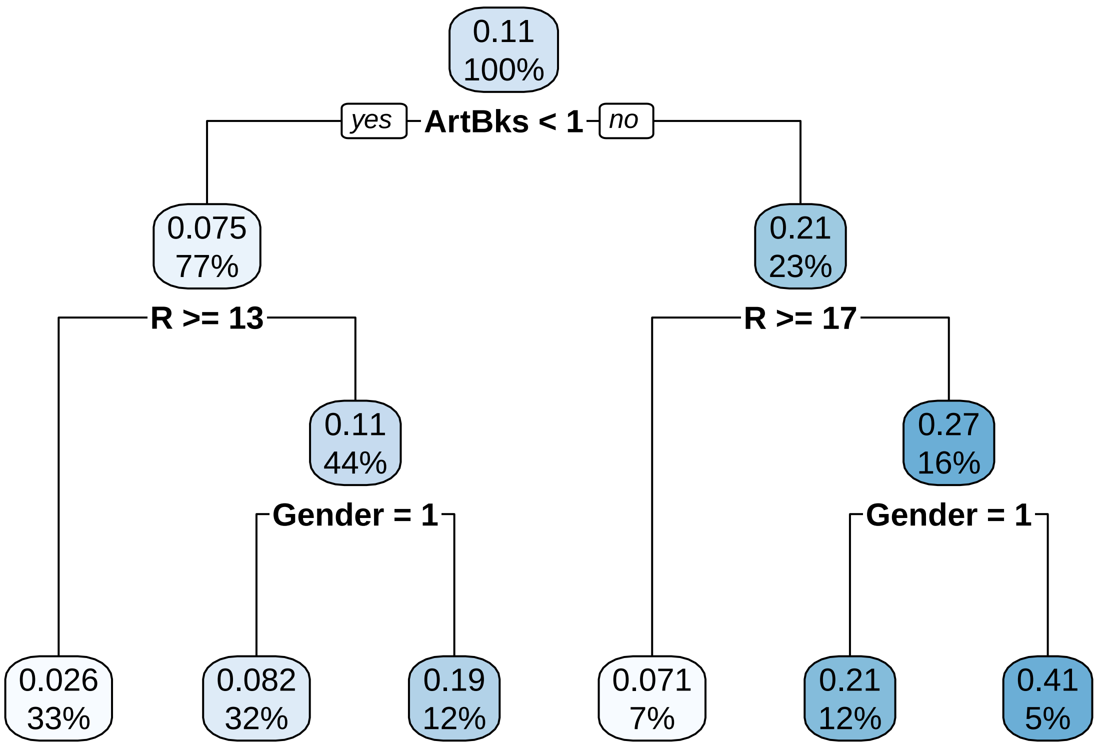
```
R_train <- cor(df.train$Florence, lmF$fitted.values)
R_train2 <- R_train**2
MSE.lmF_train <- sum(lmF$residuals**2)/nrow(df.train)
RMSE.lmF_train <- sqrt(MSE.lmF_train)

R_test <- cor(df.test$Florence, pred.lmF_test)
R_test2 <- R_test**2
MSE.lmF_test <- sum(reslmF_test**2)/nrow(df.test)
RMSE.lmF_test <- sqrt(MSE.lmF_test)
```

```
RMSE <- c(RMSE.lmF_train, RMSE.lmF_test)
R2 <- c(R_train2, R_test2)
df.lmF <- rbind.data.frame(RMSE, R2)
```

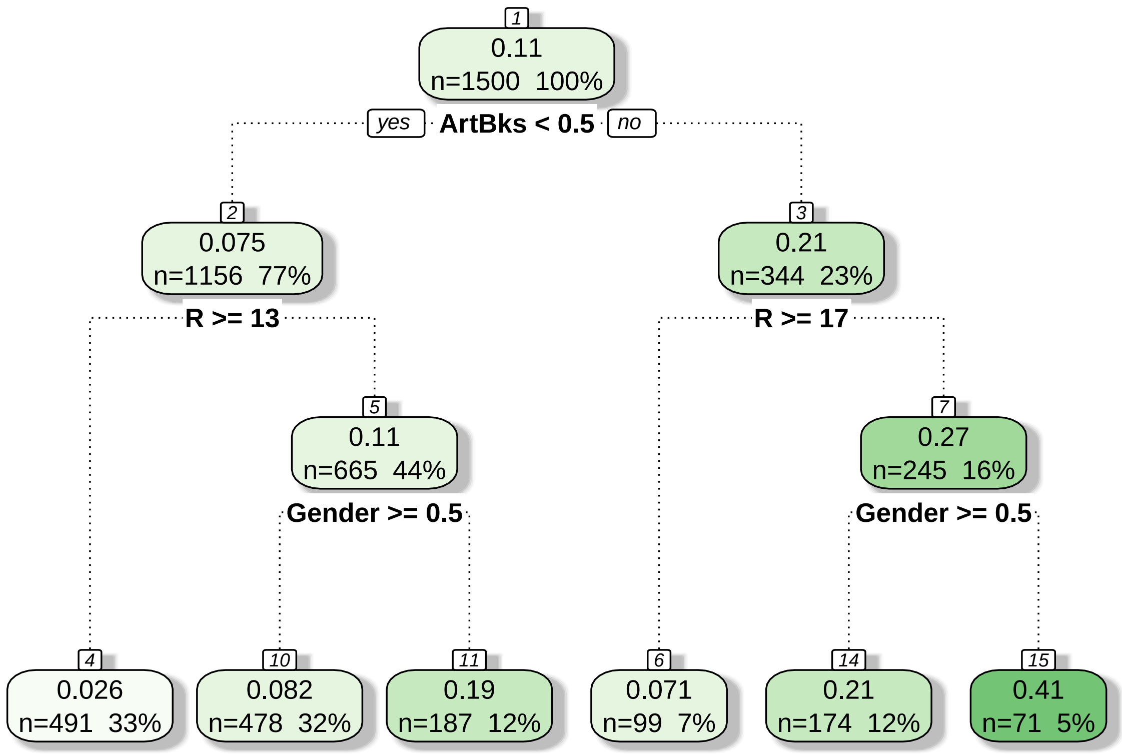
```r
colnames(df.lmF) <- c("training", "test")
rownames(df.lmF) <- c("RMSE.LM", "R_Square.LM")
df.lmF
```

```
##              training      test
## RMSE.LM     0.2884960 0.2986055
## R_Square.LM 0.1265513 0.1194512
```

```r
fit <- rpart(Florence ~ . -Seq. -ID. -FirstPurch -M -RefBks -GeogBks -ItalCook -ItalHAtlas -ItalArt,
             data = df.train, method = "anova")
rpart.plot(fit)
```




```r
fancyRpartPlot(fit)
```



Rattle 2020−Jun−24 19:18:14 Shiloh

```
pred.train_dt <- predict(fit, newdata = df.train)
pred.test_dt <- predict(fit, newdata = df.test)
MSE.train_dt <- sum((pred.train_dt - df.train$Florence)**2)/nrow(df.train)
RMSE.train_dt <- sqrt(MSE.train_dt)
MSE.test_dt <- sum((pred.test_dt - df.test$Florence)**2)/nrow(df.test)
RMSE.test_dt <- sqrt(MSE.test_dt)

r.train_dt <- cor(df.train$Florence, pred.train_dt)

RMSE_dt <- c(RMSE.train_dt, RMSE.test_dt)

r.test_dt <- cor(df.test$Florence, pred.test_dt)
r_dt <- c(r.train_dt, r.test_dt)
r2_dt <- r_dt**2

df.dt <- rbind.data.frame(RMSE_dt, r2_dt)
colnames(df.dt) <- c("training","test")
rownames(df.dt) <- c("RMSE.DT","R_Square.DT")
round(df.dt,2)
```

```
##               training test
## RMSE.DT           0.29 0.31
## R_Square.DT       0.09 0.04
```

```
df.train <- cbind.data.frame(df.train$Florence, lmF$fitted.values, pred.train_dt)
colnames(df.train) <- c("Florence_obs", "Florence_lm.pred", "Florence_dt.pred")
df.test <- cbind.data.frame(df.test$Florence, pred.lmF_test, pred.test_dt)
colnames(df.test) <- c("Florence_obs", "Florence_lm.pred", "Florence_dt.pred")
```

```
p.train1 <- ggplot(df.train, aes(x = Florence_obs)) +
  geom_point(aes(y = Florence_lm.pred), color = "red") +
  geom_point(aes(y = Florence_dt.pred), color = "steelblue") +
  xlab("Observed Median Value") +
  ylab("Predicted Median Value\nDecision Tree=blue, LM = red")

p.train1
```