Estimating Medicare Costs for all Codes

Team 02 (Aditya Dube and Richard Budden)

3/20/2023

#-#-# Predicting Medicare for All Codes #-#-#

```
  ## Step 0: Basic Setup - Install Packages / Load Libraries

  ## Step 1: Collecting Data
             part a) Import Datasets
             part b) Joining the Datasets

  ## Step 2: Explore / Prepare Data
             part a) Remove, Code, and/or Impute Data
             part b) Variable Selection
             part c) Group by State/Summarize Data

  ## Step 3: Visualization of Data
             part a) Histogram
             part b) United States Heat Map

  ## Step 4: Creating Training and Test Datasets

  ## Step 5: Build and Evaluate Linear Regression Model

  ## Step 6: Build and Evaluate CART Model

  ## Step 7: Build and Evaluate Artifical Neural Network  Model (Feedforward ANN)
```

```
## Step 0: Basic Setup - Install Packages / Load Libraries
```

```
library(dplyr)
library(tidyverse)
library(dplyr)
library(ggplot2)
library(maps)
library(caret)
library(rpart)
library(rpart.plot)
library(rattle)
library(neuralnet)
```

Step 1: Collecting Data part a) Import Datasets

```
# Load the three datasets
hospital_data <- read.csv("Hospital General Information.csv")
medicare_data <-
↪  read.csv("Medicare_Inpatient_Hospital_by_Provider_and_Service_2018_data.csv")
census_data <- read.csv("Census data.csv")
```

Initial look at the Structure of the Datasets

```
##  Structure of the dataset
str(hospital_data)
```

```
## 'data.frame':    4793 obs. of  13 variables:
##  $ Provider_ID                               : int  10001 10005 10006 10007 10008 10011 10012 10016 10
##  $ Hospital.Name                             : chr  "SOUTHEAST ALABAMA MEDICAL CENTER" "MARSHALL MEDIC
##  $ Address                                   : chr  "1108 ROSS CLARK CIRCLE" "2505 U S HIGHWAY 431 NOR
##  $ City                                      : chr  "DOTHAN" "BOAZ" "FLORENCE" "OPP" ...
##  $ State                                     : chr  "AL" "AL" "AL" "AL" ...
##  $ ZIP_Code                                  : int  36301 35957 35631 36467 36049 35235 35968 35007 35
##  $ County.Name                               : chr  "HOUSTON" "MARSHALL" "LAUDERDALE" "COVINGTON" ...
##  $ Phone.Number                              : num  3.35e+09 2.57e+09 2.57e+09 3.34e+09 3.34e+09 ...
##  $ Hospital.Type                             : chr  "Acute Care Hospitals" "Acute Care Hospitals" "Acu
##  $ Hospital.Ownership                        : chr  "Government - Hospital District or Authority" "Gov
##  $ Emergency.Services                        : chr  "Yes" "Yes" "Yes" "Yes" ...
##  $ Meets.criteria.for.meaningful.use.of.EHRs: chr  "Y" "Y" "Y" "Y" ...
##  $ Hospital.overall.rating                   : chr  "3" "2" "2" "2" ...
```

```
str(medicare_data)
```

```
## 'data.frame':    193003 obs. of  15 variables:
##  $ Provider_ID              : int  10001 10001 10001 10001 10001 10001 10001 10001 10001 10001 ...
##  $ provider_name            : chr  "Southeast Alabama Medical Center" "Southeast Alabama Medical Cent
##  $ street_address           : chr  "1108 Ross Clark Circle" "1108 Ross Clark Circle" "1108 Ross Clark
##  $ Rndrng_Prvdr_City        : chr  "Dothan" "Dothan" "Dothan" "Dothan" ...
##  $ Rndrng_Prvdr_State_Abrvtn: chr  "AL" "AL" "AL" "AL" ...
##  $ Rndrng_Prvdr_State_FIPS  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ ZIP_Code                 : int  36301 36301 36301 36301 36301 36301 36301 36301 36301 36301 ...
##  $ Rndrng_Prvdr_RUCA        : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ Rndrng_Prvdr_RUCA_Desc   : chr  "Metropolitan area core: primary flow within an urbanized area of
##  $ DRG_Cd                   : int  3 23 25 38 39 57 64 65 66 69 ...
##  $ DRG                      : chr  "\"ECMO OR TRACH W MV >96 HRS OR PDX EXC FACE, MOUTH & NECK W MAJ
##  $ Total_discharges         : int  13 33 26 11 64 30 115 107 17 53 ...
##  $ Ave_covered_charges      : num  368434 148677 118718 74449 46628 ...
##  $ Ave_total_payment        : num  81541 29062 22442 9546 6468 ...
##  $ Ave_medical_payment      : num  80435 27997 19592 7562 5073 ...
```

```
str(census_data)
```

```
## 'data.frame':    33120 obs. of  2 variables:
##  $ ZIP_Code: chr  "8600000US00601" "8600000US00602" "8600000US00603" "8600000US00606" ...
##  $ NAME    : chr  "ZCTA5 00601" "ZCTA5 00602" "ZCTA5 00603" "ZCTA5 00606" ...
```

Step 1: Collecting Data part b) Joining the Datasets

Used the dplyr package in R to join multiple datasets based on a common variable. We Joined the medicare and hospital data by Provider ID and by Zip code

```
final_data <- inner_join(medicare_data, hospital_data, by = c("Provider_ID","ZIP_Code"))
```

Step 2: Explore / Prepare Data part a) Remove, Code, and/or Impute Data

---

Data cleaning: Before analyzing the data, we need to clean it by removing missing values, fixing formatting issues, and dealing with outliers

```
# Remove rows with missing values
final_data <- final_data[complete.cases(final_data), ]
```

Step 2: Explore / Prepare Data part b) Variable Selection

Data wrangling and variable selection: We use the tidyverse package in R to manipulate and select variables from our dataset.

```
# Select relevant variables

selected_data <- final_data %>%
  select("Provider_ID", "ZIP_Code","Ave_medical_payment", "Total_discharges",
  ↪  "Ave_covered_charges", "Ave_total_payment","Hospital.Type","Hospital.Ownership",
        "Hospital.overall.rating","State", "DRG_Cd")
attach (selected_data)

# Concerting character variable to string variables
selected_data[sapply(selected_data, is.character)] <-
↪  lapply(selected_data[sapply(selected_data, is.character)],
                                      as.factor)
```

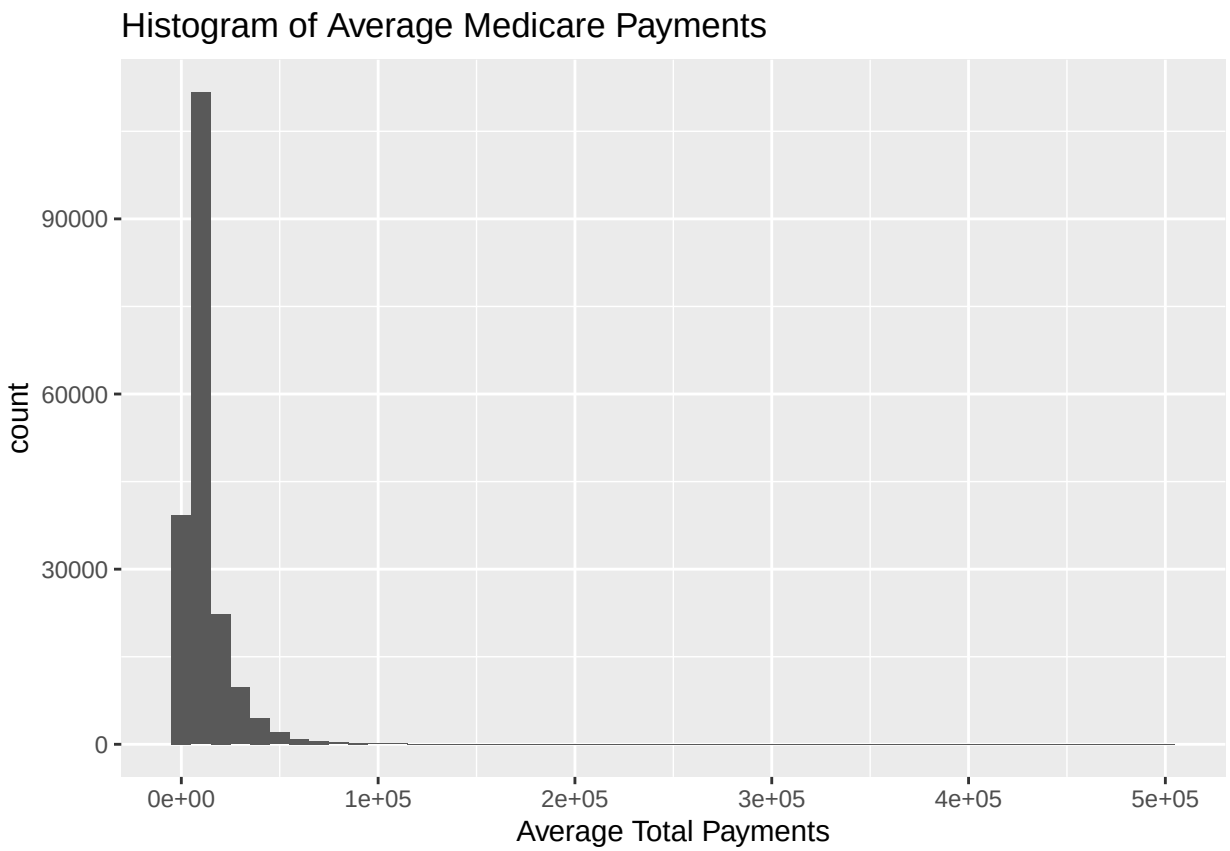
Step 2: Explore / Prepare Data part c) Group by State/Summarize Data

---

```
# Calculate the average costs per state
Avg_Mdcr_Pymt_Amt  <- selected_data %>%
  group_by(State) %>%
  summarise(Avg_Mdcr_Pymt  = Ave_medical_payment)
Avg_Mdcr_Pymt_Amt <- Avg_Mdcr_Pymt_Amt[1:15537, ]
```

Step 3: Visualization of Data part a) Histogram

```
# Create a histogram of average costs per state
# Create a histogram of average costs per state
ggplot(selected_data , aes(x = Ave_medical_payment)) +
  geom_histogram(binwidth = 10000) +
  labs(title = "Histogram of Average Medicare Payments", x = "Average Total Payments")
```



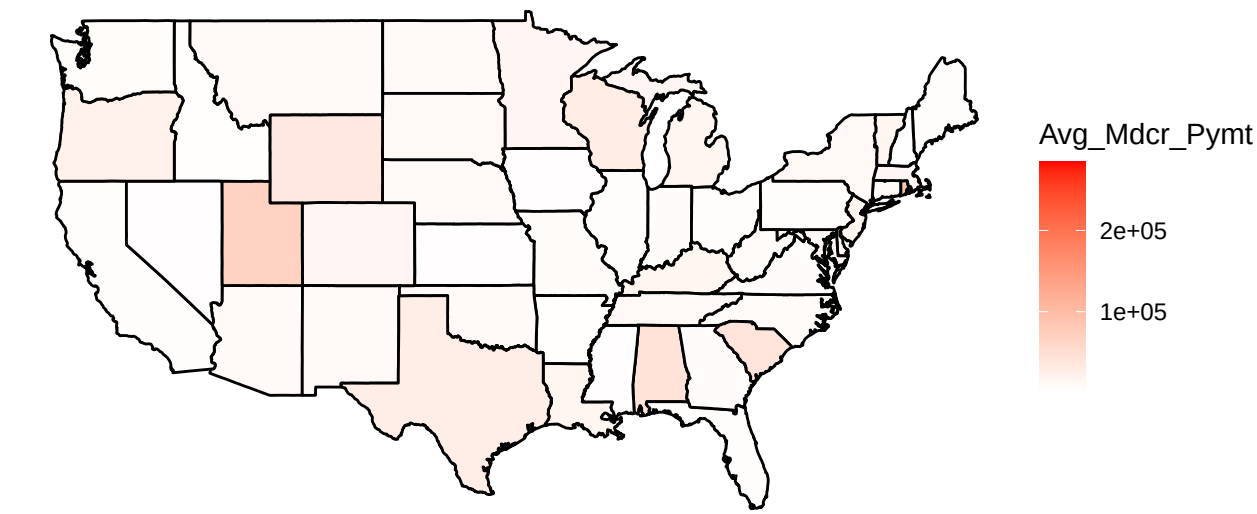
Step 3: Visualization of Data part b) United States Heat Map

```
# Create a US density map of average costs per state
us_map <- map_data("state")
usmap <- cbind(us_map,Avg_Mdcr_Pymt_Amt)

head(usmap)
```

```
##        long      lat group order  region subregion State Avg_Mdcr_Pymt
## 1 -87.46201 30.38968     1     1 alabama      <NA>    AK     43056.846
## 2 -87.48493 30.37249     1     2 alabama      <NA>    AK     12839.000
## 3 -87.52503 30.37249     1     3 alabama      <NA>    AK     10311.083
## 4 -87.53076 30.33239     1     4 alabama      <NA>    AK     13333.692
## 5 -87.57087 30.32665     1     5 alabama      <NA>    AK     17988.595
## 6 -87.58806 30.32665     1     6 alabama      <NA>    AK      9322.145
```

```
ggplot(usmap, aes(x = long, y = lat, group = group, fill = Avg_Mdcr_Pymt)) +
  geom_polygon(color = "black") +
  scale_fill_gradient(low = "white", high = "red", na.value = "lightgrey") +
  theme_void() + coord_map() +
  labs(title = "Average Medicare Payments by State (All Codes)")
```



Step 4: Creating Training and Test Datasets

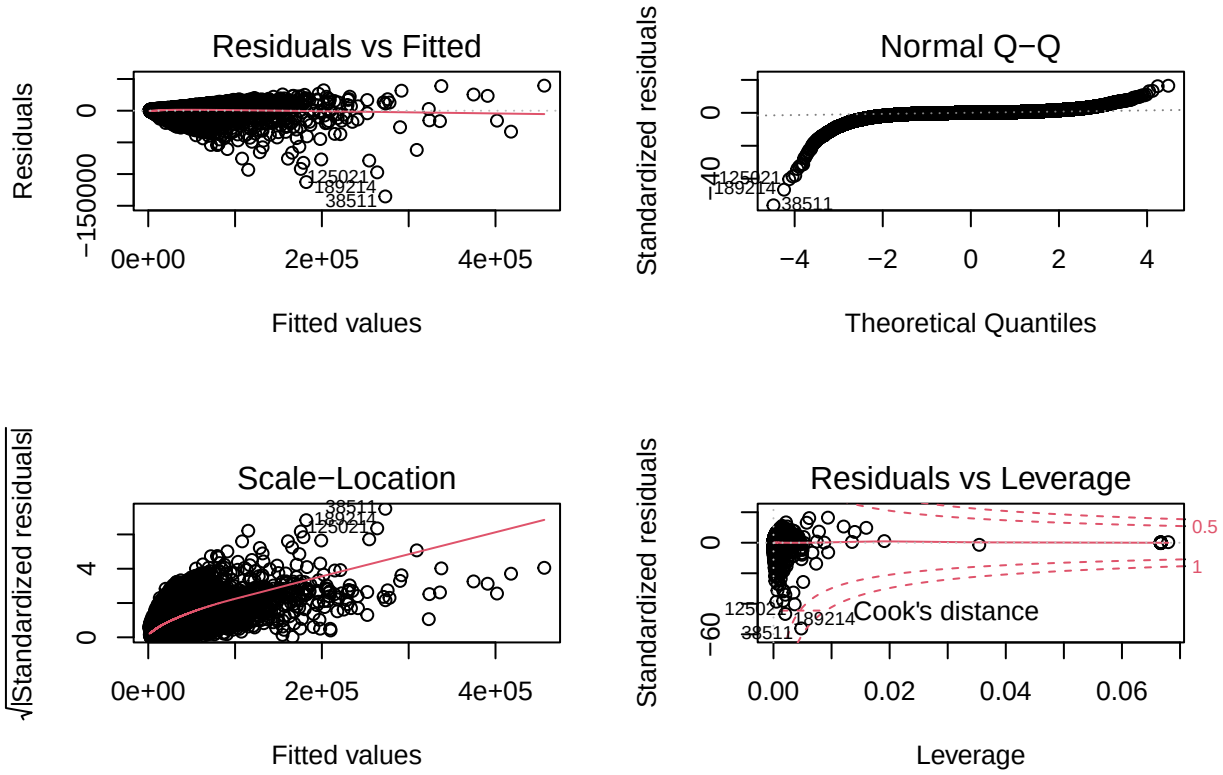
```r
# Split the data into training and testing sets
set.seed(123)
library(caret)
trainIndex <- createDataPartition(selected_data$Ave_medical_payment, p = 0.7, list =
↪  FALSE)
train <- selected_data[trainIndex, ]
test <- selected_data[-trainIndex, ]
```

Step 5: Build and Evaluate Linear Regression Model

```r
# Build the linear regression model
model <- lm(Ave_medical_payment ~ Total_discharges + Ave_covered_charges +
↪  Ave_total_payment+ Hospital.overall.rating+Hospital.Ownership, data = train)

# Predict on the testing set
predictions <- predict(model, newdata = test)


# Check for the model assumptions
par(mfrow = c(2, 2))
plot(model)
```



Evaluate the performance of the Linear Model

```
RMSE(predictions, test$Ave_medical_payment)
```

```
## [1] 2707.048
```

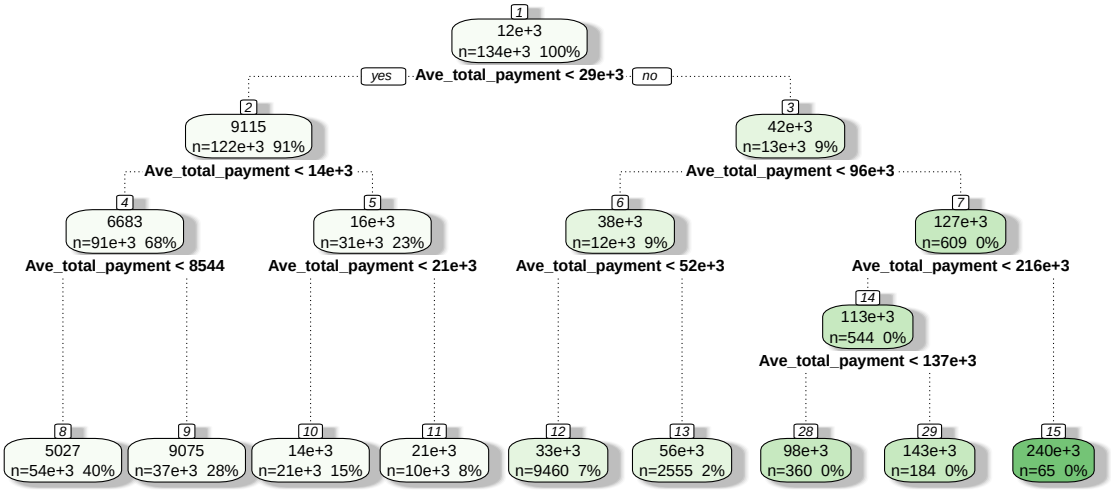
```
R2(predictions, test$Ave_medical_payment)
```

```
## [1] 0.9615135
```

Step 6 Build and Evaluate CART Model (Regression Tree)

```
# Build the regression tree model
model2 <- rpart(Ave_medical_payment ~ Total_discharges + Ave_covered_charges +
↪  Ave_total_payment+ Hospital.overall.rating+Hospital.Ownership, data = train, method =
↪  "anova")

#Plot
#install.packages("RGtk2")
# Plot the tree
fancyRpartPlot(model2)
```



Rattle 2023−Apr−21 22:07:20 richardbudden

```
# Predict on the testing set
predictions <- predict(model2, newdata = test)

#Model 2
RMSE(predictions, test$Ave_medical_payment)
```

```
## [1] 3732.107
```

```
R2(predictions, test$Ave_medical_payment)
```

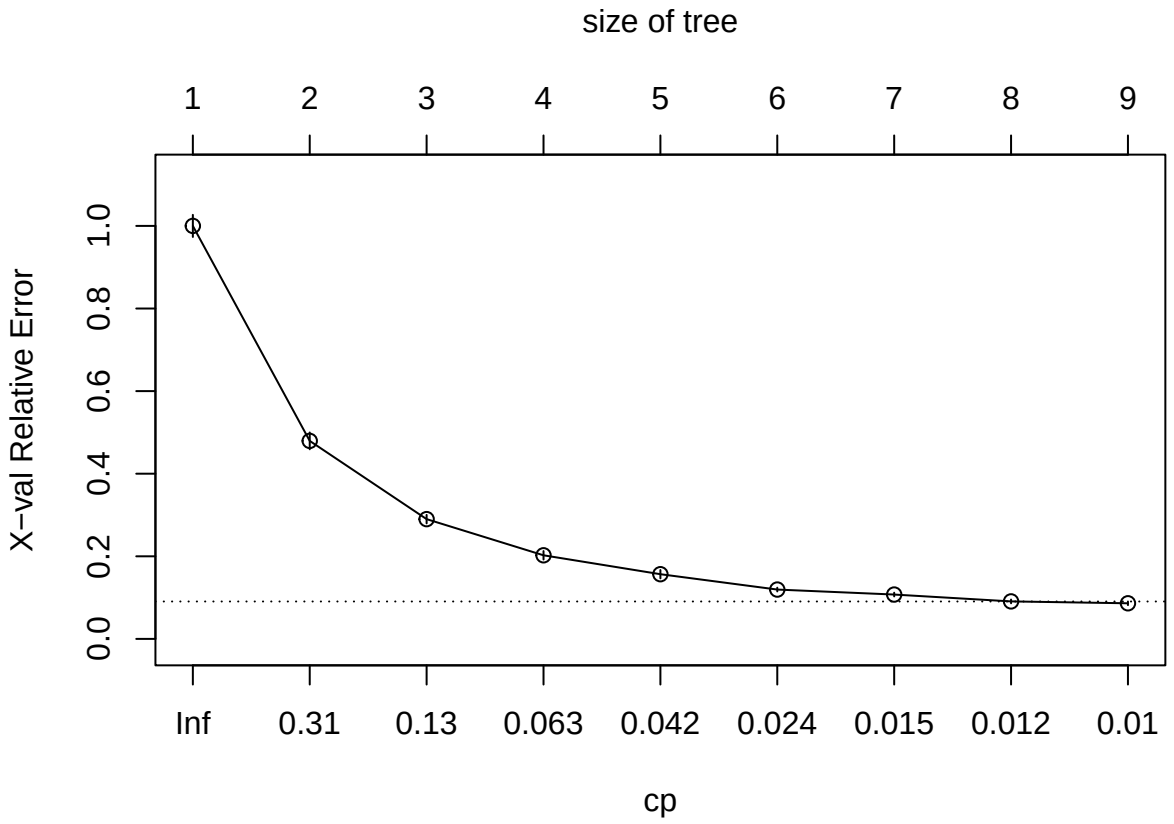
```
## [1] 0.9266838
```

```
printcp(model2)
```

```
##
## Regression tree:
## rpart(formula = Ave_medical_payment ~ Total_discharges + Ave_covered_charges +
##     Ave_total_payment + Hospital.overall.rating + Hospital.Ownership,
##     data = train, method = "anova")
##
## Variables actually used in tree construction:
## [1] Ave_total_payment
##
```

```
## Root node error: 2.3983e+13/134469 = 178354745
##
## n= 134469
##
##         CP nsplit rel error   xerror      xstd
## 1 0.520837      0  1.000000 1.000016 0.0267865
## 2 0.190373      1  0.479163 0.479461 0.0203668
## 3 0.088440      2  0.288790 0.289994 0.0102724
## 4 0.045560      3  0.200350 0.202415 0.0102608
## 5 0.039074      4  0.154790 0.156628 0.0101509
## 6 0.015027      5  0.115716 0.119450 0.0045184
## 7 0.014344      6  0.100688 0.107396 0.0045196
## 8 0.010235      7  0.086344 0.090775 0.0045387
## 9 0.010000      8  0.076109 0.086184 0.0044308
```

```
plotcp(model2)
```




```
pfit = prune(model2, cp=model2$ptable[which.min(model2$cptable[,"xerror"]), "CP"])
pfit
```

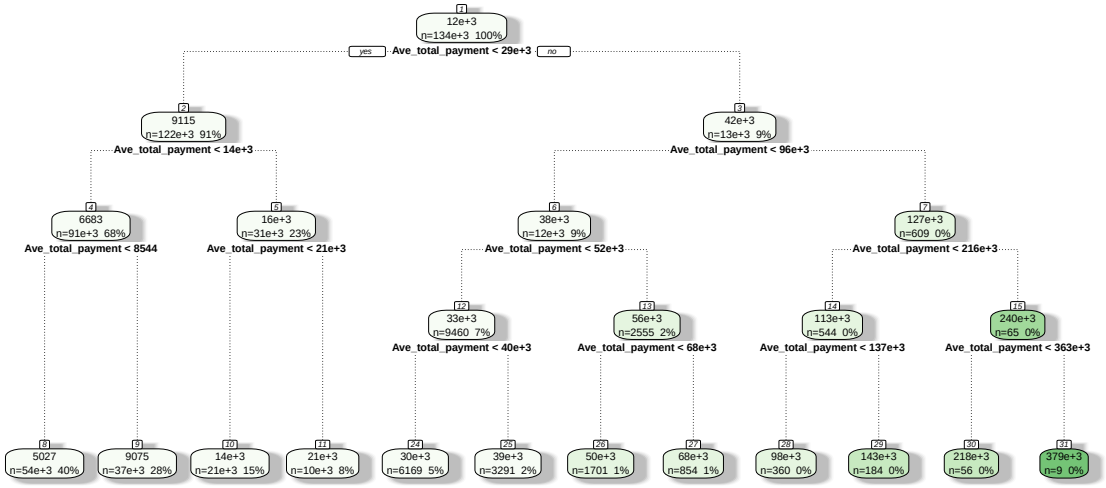
```
## n= 134469
##
## node), split, n, deviance, yval
##       * denotes terminal node
```

```
##
##  1) root 134469 2.398318e+13  12217.130
##    2) Ave_total_payment< 29423.81 121845 3.255134e+12   9114.798
##      4) Ave_total_payment< 13688.68 90958 5.359009e+11   6683.480
##        8) Ave_total_payment< 8544.083 53735 7.822789e+10   5026.764 *
##        9) Ave_total_payment>=8544.083 37223 9.727458e+10   9075.109 *
##      5) Ave_total_payment>=13688.68 30887 5.981552e+11  16274.700
##       10) Ave_total_payment< 20575.88 20574 1.210825e+11  13911.870 *
##       11) Ave_total_payment>=20575.88 10313 1.330592e+11  20988.450 *
##    3) Ave_total_payment>=29423.81 12624 8.236729e+12  42160.340
##      6) Ave_total_payment< 95752.02 12015 1.933368e+12  37878.760
##       12) Ave_total_payment< 51825.62 9460 4.224816e+11  32922.730 *
##       13) Ave_total_payment>=51825.62 2555 4.182088e+11  56228.680 *
##      7) Ave_total_payment>=95752.02 609 1.737602e+12 126632.000
##       14) Ave_total_payment< 215964.7 544 4.910493e+11 113072.500
##         28) Ave_total_payment< 137279.9 360 1.107241e+11  97885.950 *
##         29) Ave_total_payment>=137279.9 184 1.348548e+11 142785.200 *
##       15) Ave_total_payment>=215964.7 65 3.094308e+11 240115.400 *
```

```r
# Build the regression tree model
control_setting <- rpart.control(minsplit = 2, cp = .005, xval = 10)

model2 <- rpart(Ave_medical_payment ~ Total_discharges + Ave_covered_charges +
↪  Ave_total_payment+ Hospital.overall.rating+Hospital.Ownership, data = train, method =
↪  "anova", control = control_setting)

# Plot
#install.packages("RGtk2")
fancyRpartPlot(model2)
```



Rattle 2023−Apr−21 22:07:21 richardbudden

```
# Predict on the testing set
predictions <- predict(model2, newdata = test)

RMSE(predictions, test$Ave_medical_payment)
```

```
## [1] 3309.706
```

```
R2(predictions, test$Ave_medical_payment)
```

```
## [1] 0.9422925
```

Evaluate the performance of the CART model

```
#Model 2
RMSE(predictions, test$Ave_medical_payment)
```

```
## [1] 3309.706
```

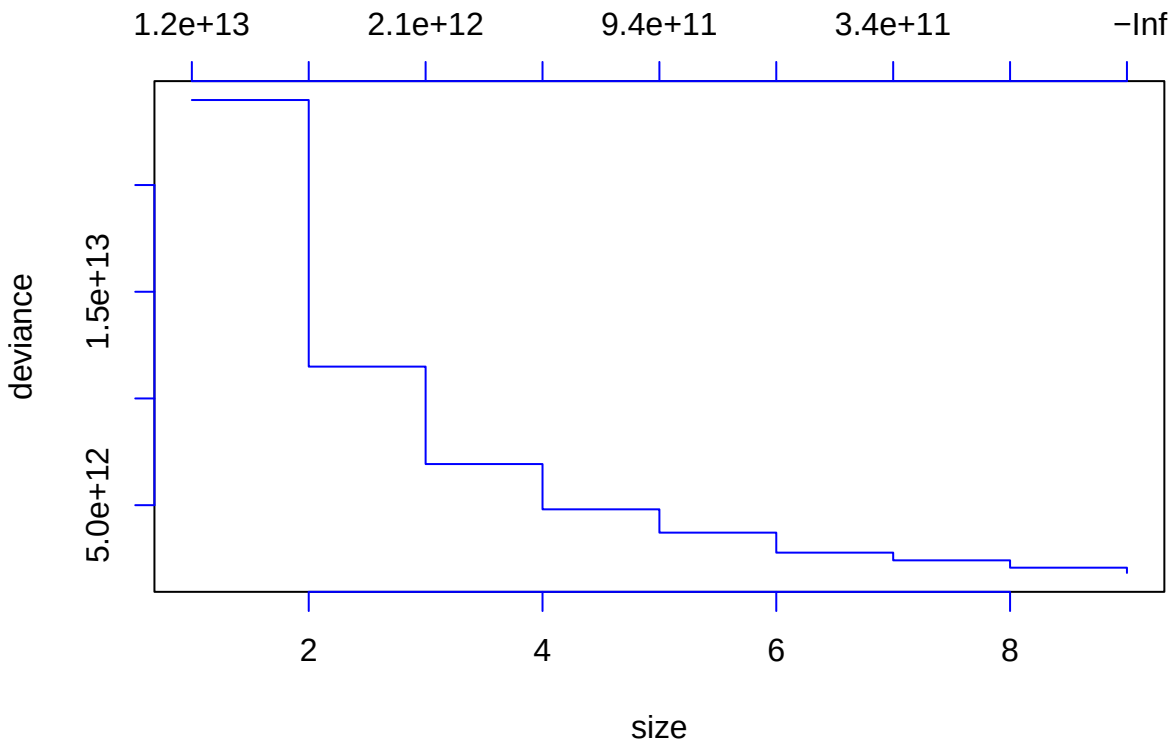
```
R2(predictions, test$Ave_medical_payment)
```

```
## [1] 0.9422925
```

Data pruning for Regression Tree

```
# Prune the tree, display pruned tree
library(tree)

tree.model <- tree(Ave_medical_payment ~ Total_discharges + Ave_covered_charges +
↪  Ave_total_payment+ Hospital.overall.rating+Hospital.Ownership, data = train)
prune.data <- prune.tree(tree.model)
plot(prune.data, col = "blue")
```



```
## Step 7 : Build and Evaluate Artifical Neural Network  Model (Feedforward ANN)
```

Feedforward ANN

```
## Scaling
scale_data <- final_data%>%
  select("Ave_medical_payment", "Total_discharges", "Ave_covered_charges",
  ↪  "Ave_total_payment")

set.seed(123)
```

```
trainIndex2 <- createDataPartition(scale_data$Ave_medical_payment, p = 0.7, list = FALSE)
train2 <- scale_data[trainIndex2, ]
test2 <- scale_data[-trainIndex2, ]
```

Min-Max Normalization and Scaling the input variable

```
#transform your factor to numeric.
#tranform it to a factor and the to numeric
selected_data$Hospital.Ownership <- as.numeric(as.factor(Hospital.Ownership))
selected_data$Hospital.overall.rating <- as.numeric(as.factor(Hospital.overall.rating))
```

```
selected_data$Ave_medical_payment <- (selected_data$Ave_medical_payment -
↪  min(selected_data$Ave_medical_payment)) / (max(selected_data$Ave_medical_payment) -
↪  min(selected_data$Ave_medical_payment))

selected_data$Total_discharges <- (selected_data$Total_discharges -
↪  min(selected_data$Total_discharges)) / (max(selected_data$Total_discharges) -
↪  min(selected_data$Total_discharges))

selected_data$Ave_covered_charges <- (selected_data$Ave_covered_charges -
↪  min(selected_data$Ave_covered_charges)) / (max(selected_data$Ave_covered_charges) -
↪  min(selected_data$Ave_covered_charges))

 selected_data$Ave_total_payment <- (selected_data$Ave_total_payment -
↪  min(selected_data$Ave_total_payment)) / (max(selected_data$Ave_total_payment) -
↪  min(selected_data$Ave_total_payment))

selected_data$Hospital.overall.rating <- (selected_data$Hospital.overall.rating -
↪  min(selected_data$Hospital.overall.rating)) /
↪  (max(selected_data$Hospital.overall.rating) -
↪  min(selected_data$Hospital.overall.rating))

selected_data$Hospital.Ownership <- (selected_data$Hospital.Ownership -
↪  min(selected_data$Hospital.Ownership)) / (max(selected_data$Hospital.Ownership) -
↪  min(selected_data$Hospital.Ownership))


set.seed(123)
inp <- sample(2, nrow(selected_data), replace = TRUE, prob = c(0.7, 0.3))
training_data <- selected_data[inp==1, ]
test_data <- selected_data[inp==2, ]
```

```
#from RBloggers "Selecting the number of neurons in the hidden layer of a neural network"
#A Variation of this rule suggests to choose a number of hidden neurons between one and
↪  the number of Inputs minus the number of outputs

#Upon our Model with 5 Inputs and 1 Output we set Hidden Levels at 4 (5-1=4)


set.seed(333)
model3 <- neuralnet(Ave_medical_payment ~ Total_discharges + Ave_covered_charges +
↪  Ave_total_payment+Hospital.Ownership+ Hospital.overall.rating,
```

```
                data = training_data,
                hidden = 4,
                linear.output = FALSE)

# Predict on the testing set
predictions <- predict(model3, test_data)

# Evaluate the performance of the model
RMSE(predictions, test_data$Ave_medical_payment)
```

```
## [1] 0.005268131
```

```
R2(predictions, test_data$Ave_medical_payment)
```

```
##           [,1]
## [1,] 0.9623551
```

```
# plot neural network
plot(model3, rep = "best")
```