Wielowymiarowa Analiza Danych

Adam Matuszczyk

28 06 2019

Wstęp

#Nieprzypadkowo wybrałem dane, których użyłem do zaliczenia przedmiotu na MSAD 2018/2019 Analiza współzależności zjawisk. Analiza regresji liniowej rządzi się swoimi bardzo restrykcyjnymi warunkami, których spełnienie w tym wypadku było bardzo trudne do spełnienia. Chciałem jak drzewko, które samo reguluje warunki i inne elementy poradzi sobie z tym problemem. Dane pochodzą ze strony: https://www.kaggle.com/mirichoi0218/insurance#

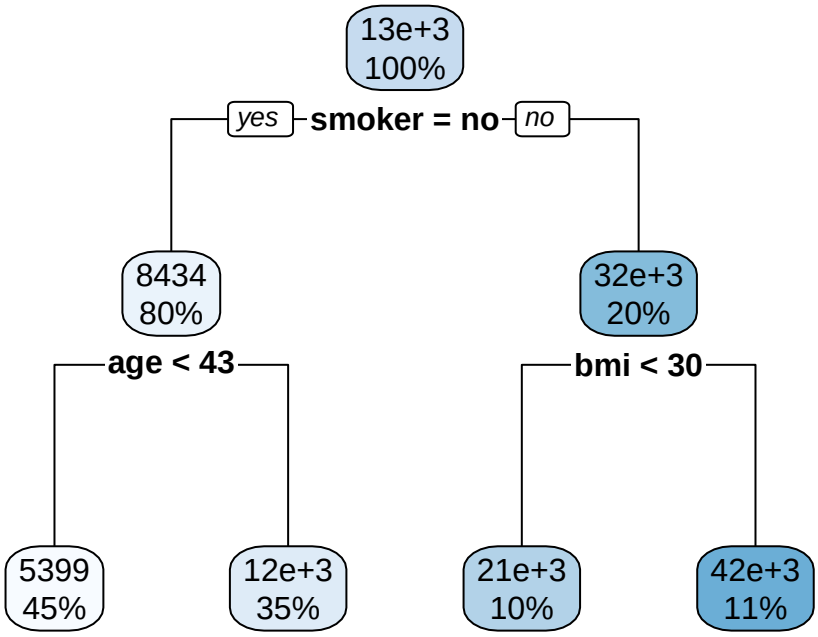
```
library("tidyverse")
library("rpart.plot")
library("knitr")
```

```
d <- read_csv("insurance.csv")
d <- d %>%
  mutate_if(is.character, as.factor)
summary(d)
```

```
##       age            sex           bmi           children     smoker
##  Min.   :18.00   female:662   Min.   :15.96   Min.   :0.000   no :1064
##  1st Qu.:27.00   male  :676   1st Qu.:26.30   1st Qu.:0.000   yes: 274
##  Median :39.00                Median :30.40   Median :1.000
##  Mean   :39.21                Mean   :30.66   Mean   :1.095
##  3rd Qu.:51.00                3rd Qu.:34.69   3rd Qu.:2.000
##  Max.   :64.00                Max.   :53.13   Max.   :5.000
##        region       charges
##  northeast:324   Min.   : 1122
##  northwest:325   1st Qu.: 4740
##  southeast:364   Median : 9382
##  southwest:325   Mean   :13270
##                  3rd Qu.:16640
##                  Max.   :63770
```

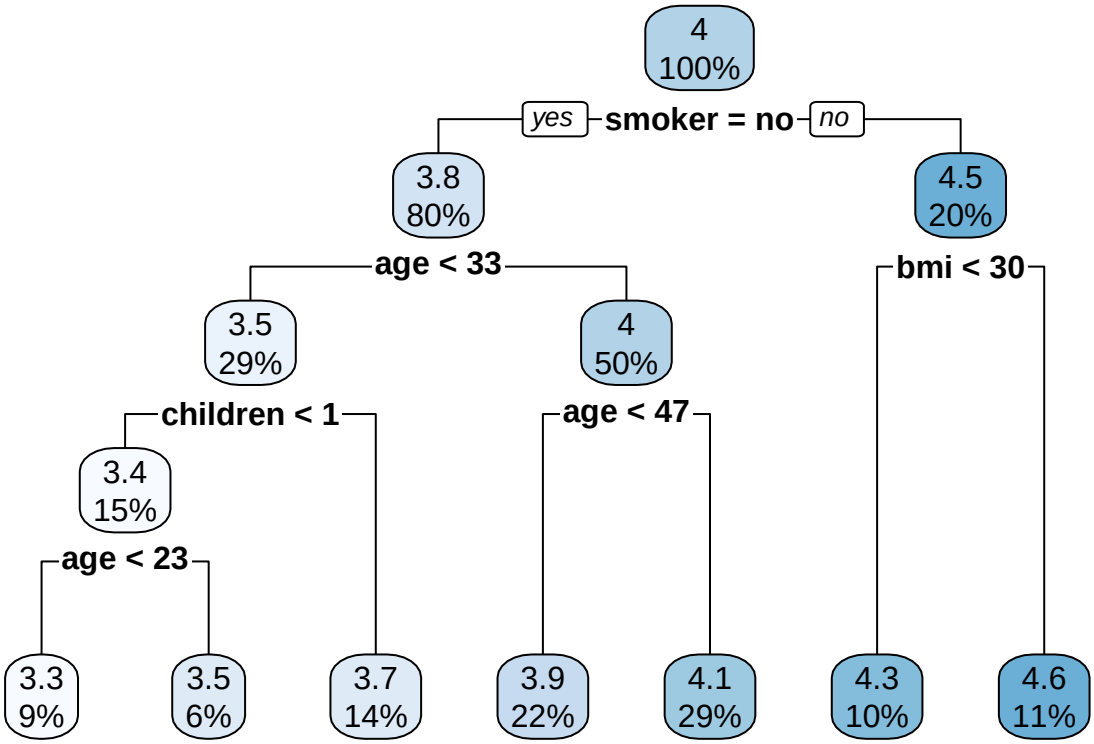
1.0 Budowa drzewka regresyjnego

```
m_tree <-  rpart(charges ~ .,d)
#m_tree
rpart.plot(m_tree)
```



Ze względu na nie dość czytelne i trudne do zinterpretowania informacje z węzłów z funkcja wykładniczą postanowiłem przedstawiac w dalszej analizie zmienną objaśnianą w postaci *log10*.

```
m_tree <-  rpart(log10(charges) ~ .,d) #zmienna wyjaśniana może być logarytmowana
#m_tree
rpart.plot(m_tree)
```



Z naszego drzewka mozna wyciagnąć wstepne dane, że *10^4* wynoszą średnie koszta leczenia gdy nie masz innych danych i chciałbys poprzestać tylko na informacji dla konkretnego klienta.Najważniejszą informacją z pierwszego węzła jest nałóg palania papierosów. Na tym poziomie widać również, że jeśli nie palisz to bmi nie ma znaczenia, a jeśli jesteś palący ma to znaczenie dla dalszych kosztów leczenia. Dla niepalących istotną informacją jest wiek, że jeśći jesteś młodszy niż 33lata to średnia kosztów leczenia wynosi *10^3,5*.

2.0 Sprawdzenie stopnia skomplikowania drzewka

2.1 Sprawdzenie podziału drzewka przy danych domyślnych

```
printcp(m_tree)
```

```
##
## Regression tree:
## rpart(formula = log10(charges) ~ ., data = d)
##
## Variables actually used in tree construction:
## [1] age      bmi      children smoker
##
## Root node error: 213.22/1338 = 0.15936
##
## n= 1338
##
##         CP nsplit rel error  xerror     xstd
```

```
## 1 0.442898      0   1.00000 1.00208 0.031939
## 2 0.254962      1   0.55710 0.55834 0.020305
## 3 0.041125      2   0.30214 0.31046 0.018073
## 4 0.032810      3   0.26102 0.26319 0.017919
## 5 0.028337      4   0.22821 0.24968 0.017423
## 6 0.012461      5   0.19987 0.21087 0.017514
## 7 0.010000      6   0.18741 0.19553 0.016540
```

```
min(m_tree$cptable[,"xerror"])
```

```
## [1] 0.1955274
```

Z danych domyślnych wynika, że najniższą wartość błędu poprawności krzyżowej *(xerror)* mamy przy wartości parametru złożoności *(cp)* na poziomie 0,01, dlatego przystapię do dalszej analizy podziału drzewka ze zmienianą wartością cp = 0.0001

2.2 Podział drzewka przy wartości cp= 0,0001

```
#?rpart.control
set.seed(1)
m_tree <-  rpart(log10(charges) ~ .,d, control = rpart.control(cp = 0.0001))
#rpart.plot(m_tree)
printcp(m_tree)
```

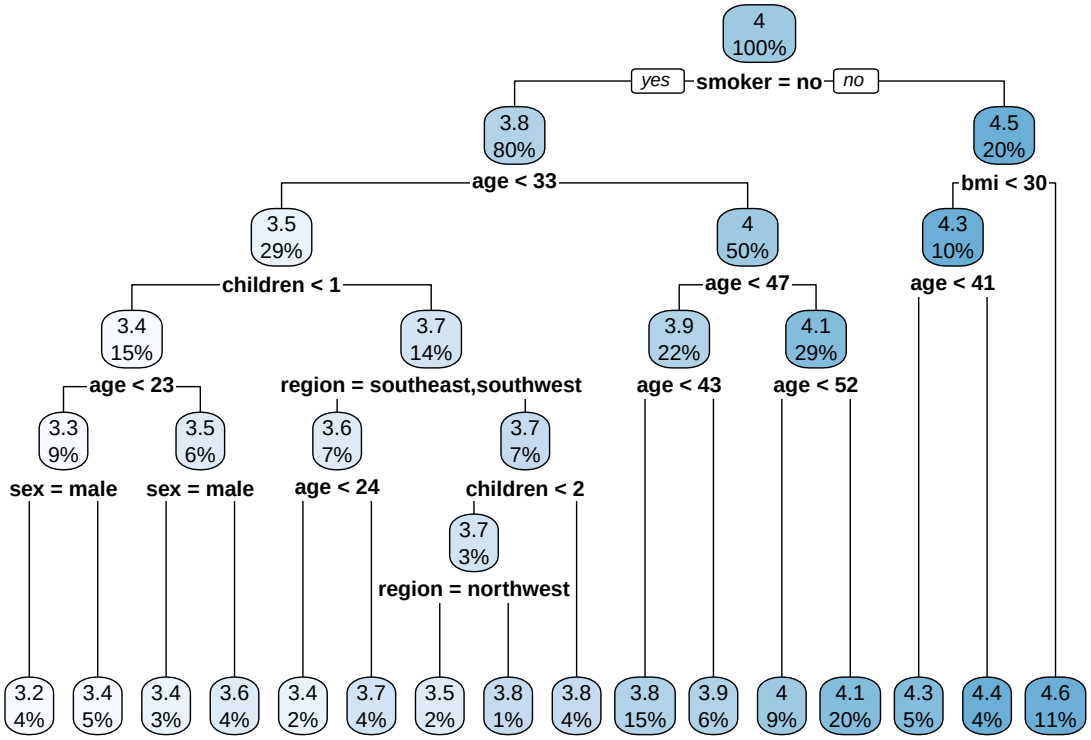
```
##
## Regression tree:
## rpart(formula = log10(charges) ~ ., data = d, control = rpart.control(cp = 1e-04))
##
## Variables actually used in tree construction:
## [1] age      bmi      children region   sex      smoker
##
## Root node error: 213.22/1338 = 0.15936
##
## n= 1338
##
##            CP nsplit rel error  xerror     xstd
## 1  0.44289785      0   1.00000 1.00301 0.031967
## 2  0.25496176      1   0.55710 0.55868 0.020309
## 3  0.04112461      2   0.30214 0.31237 0.018099
## 4  0.03280979      3   0.26102 0.26699 0.017764
## 5  0.02833662      4   0.22821 0.25780 0.018599
## 6  0.01246139      5   0.19987 0.21817 0.018296
## 7  0.00528835      6   0.18741 0.20034 0.017385
## 8  0.00479175      7   0.18212 0.19518 0.017388
## 9  0.00444849      8   0.17733 0.19390 0.017580
## 10 0.00434874      9   0.17288 0.19043 0.017389
## 11 0.00276466     10   0.16853 0.18537 0.017127
## 12 0.00267414     12   0.16300 0.18405 0.017540
## 13 0.00210557     13   0.16033 0.18342 0.017758
## 14 0.00200919     14   0.15822 0.18456 0.018216
## 15 0.00199696     15   0.15621 0.18448 0.018668
## 16 0.00169988     16   0.15422 0.18503 0.018882
## 17 0.00150340     17   0.15252 0.18357 0.019063
```

```
## 18 0.00144601     18   0.15101 0.18256 0.019033
## 19 0.00135736     19   0.14957 0.18216 0.019098
## 20 0.00135637     20   0.14821 0.18116 0.018903
## 21 0.00130694     23   0.14414 0.18052 0.018906
## 22 0.00118269     24   0.14283 0.18002 0.018913
## 23 0.00106361     25   0.14165 0.18074 0.019155
## 24 0.00103429     26   0.14059 0.18013 0.019007
## 25 0.00096877     28   0.13852 0.18065 0.019002
## 26 0.00078513     29   0.13755 0.18096 0.019041
## 27 0.00073828     30   0.13676 0.18270 0.019292
## 28 0.00072744     31   0.13603 0.18463 0.019272
## 29 0.00067238     32   0.13530 0.18412 0.019228
## 30 0.00067231     33   0.13463 0.18387 0.019228
## 31 0.00063895     35   0.13328 0.18384 0.019193
## 32 0.00053250     37   0.13200 0.18381 0.019159
## 33 0.00049475     38   0.13147 0.18252 0.019119
## 34 0.00047357     39   0.13098 0.18302 0.019160
## 35 0.00045970     40   0.13050 0.18322 0.019164
## 36 0.00045491     41   0.13004 0.18270 0.019109
## 37 0.00041706     42   0.12959 0.18388 0.019198
## 38 0.00040508     43   0.12917 0.18513 0.019338
## 39 0.00038075     44   0.12877 0.18533 0.019321
## 40 0.00035034     45   0.12839 0.18641 0.019401
## 41 0.00034455     47   0.12768 0.18609 0.019358
## 42 0.00032362     48   0.12734 0.18702 0.019508
## 43 0.00031765     49   0.12702 0.18710 0.019508
## 44 0.00030471     54   0.12543 0.18675 0.019470
## 45 0.00030213     55   0.12512 0.18660 0.019472
## 46 0.00026802     56   0.12482 0.18756 0.019508
## 47 0.00023989     57   0.12455 0.18757 0.019530
## 48 0.00022917     60   0.12383 0.18792 0.019532
## 49 0.00020628     63   0.12315 0.18778 0.019525
## 50 0.00019300     64   0.12294 0.18792 0.019557
## 51 0.00017437     65   0.12275 0.18800 0.019582
## 52 0.00015119     68   0.12222 0.18832 0.019543
## 53 0.00015029     69   0.12207 0.18855 0.019539
## 54 0.00013674     70   0.12192 0.18821 0.019526
## 55 0.00013509     71   0.12179 0.18811 0.019524
## 56 0.00013189     72   0.12165 0.18810 0.019525
## 57 0.00012311     73   0.12152 0.18845 0.019557
## 58 0.00011900     74   0.12140 0.18823 0.019553
## 59 0.00011817     75   0.12128 0.18823 0.019553
## 60 0.00011549     76   0.12116 0.18834 0.019552
## 61 0.00010759     77   0.12104 0.18838 0.019574
## 62 0.00010502     78   0.12093 0.18844 0.019575
## 63 0.00010201     79   0.12083 0.18832 0.019575
## 64 0.00010072     80   0.12073 0.18839 0.019576
## 65 0.00010000     81   0.12063 0.18840 0.019576
```

przycinamy drzewko dla parametru *cp = 0,002* ze względu na najniższą wartość błędu poprawności krzyżowej *xerror*. poniżej tej wartości dochodzi do tzv zjawiska *overfitted*.

2.3 Przycinam drzewko przy wartości cp= 0,002

```
m_tree <- prune(m_tree, cp = 0.002)
rpart.plot(m_tree)
```



Mamy ciekawe wnioski i dwa narzucające sie modele postepowania. Czy lepiej byc palącym i przy okazji mieć bmi>=30? Wtedy praktycznie nie spełniając żadnych dodatkowych warunków przy średnich kosztach leczenia na poziomie 10^4,6 możemy sobie żyć w spokoju, ale chyba szczęsliwi przynajmniej o smak np. smalców lub boczków na pszennym pieczywie. Czy być mężczyzną na dododatek niezbyt starym bo młodszym niż.23lata jeszcze bez dzieci i nie palić!!! Zostawiam do przemyslenia.

Zauważmy, że zmienna *age* wielokrotnie wchodzi w interakcje w różnych węzłach i w wielu wypadkach w gałęziach przy różnych innych parametrach jest istotna

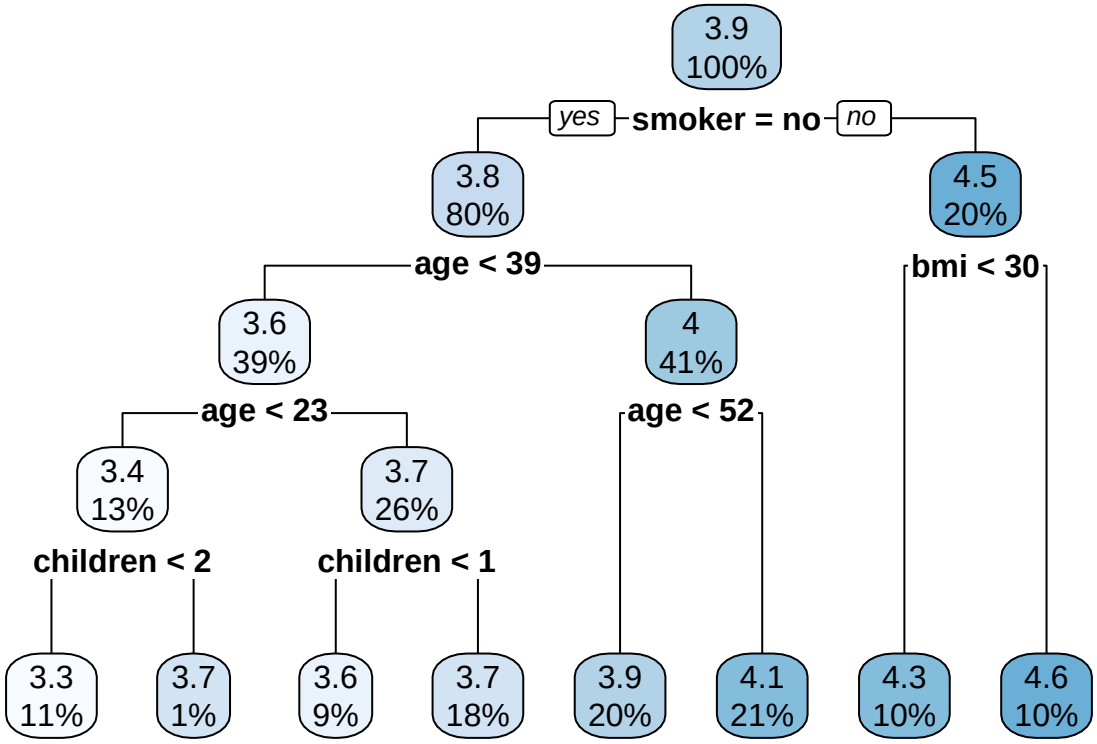
3.0 Sprawdzane dokładności prognozy

Aby zbadać sprawdzalność drzewka utworzę z danych dwa zbiory *train* oraz *test*. Do zbioru *train* zostanie wylosowanych 70%danych, aby każdowo otrzymywać podobny zbiór użyłem funkcji *set.seed()*.

```
#budowa modelu treningowego i testowego
set.seed(1)
ind <- sample(1:nrow(d), 0.7*nrow(d))
train <- slice(d, ind)
test <- slice(d, -ind)
```

wykonujemy analizę dla zbioru treningowego

```
m_tree <- rpart(log10(charges) ~ ., train)
#m_tree
rpart.plot(m_tree)
```




```
printcp(m_tree)
```

```
##
## Regression tree:
## rpart(formula = log10(charges) ~ ., data = train)
##
## Variables actually used in tree construction:
## [1] age      bmi      children smoker
##
## Root node error: 145.99/936 = 0.15598
##
## n= 936
##
##         CP nsplit rel error  xerror     xstd
## 1 0.451013      0   1.00000 1.00294 0.038570
## 2 0.254752      1   0.54899 0.55110 0.023942
## 3 0.055009      2   0.29423 0.30059 0.019287
## 4 0.028667      3   0.23923 0.25375 0.021922
## 5 0.020190      4   0.21056 0.21749 0.021495
## 6 0.014094      5   0.19037 0.19919 0.021222
## 7 0.013254      6   0.17628 0.19676 0.021593
```

```
## 8 0.010000      7   0.16302 0.18869 0.020848
```

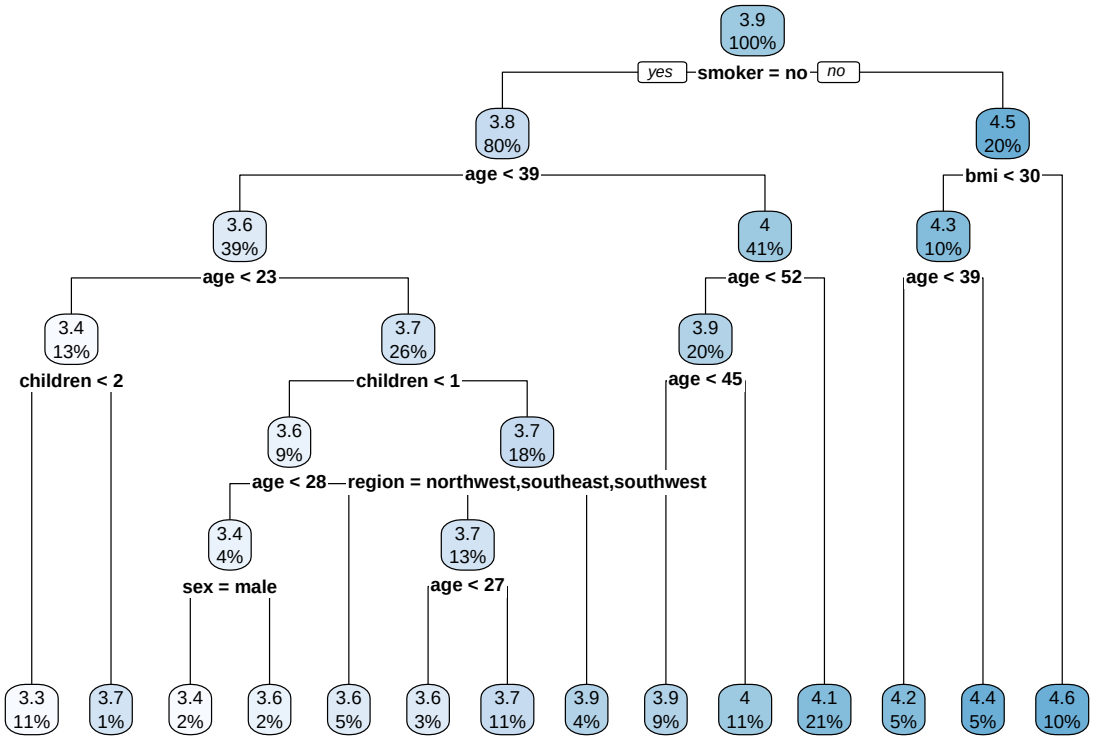
dopasowujemy wartości *cp* w celu optymalizacji drzewka zbioru treningowego

```
m_tree <- rpart(log10(charges) ~ ., train, control = rpart.control(cp = 0.0001))
#m_tree
#rpart.plot(m_tree)
#?rpart.control
printcp(m_tree)
```

```
##
## Regression tree:
## rpart(formula = log10(charges) ~ ., data = train, control = rpart.control(cp = 1e-04))
##
## Variables actually used in tree construction:
## [1] age      bmi      children region   sex      smoker
##
## Root node error: 145.99/936 = 0.15598
##
## n= 936
##
##            CP nsplit rel error  xerror     xstd
## 1  0.45101311      0   1.00000 1.00468 0.038572
## 2  0.25475230      1   0.54899 0.55125 0.023914
## 3  0.05500860      2   0.29423 0.30490 0.019778
## 4  0.02866659      3   0.23923 0.25388 0.022031
## 5  0.02018976      4   0.21056 0.22414 0.022197
## 6  0.01409411      5   0.19037 0.20212 0.020974
## 7  0.01325355      6   0.17628 0.19881 0.021167
## 8  0.00570527      7   0.16302 0.18966 0.020960
## 9  0.00539593      8   0.15732 0.17810 0.019227
## 10 0.00481367      9   0.15192 0.17378 0.019114
## 11 0.00397360     10   0.14711 0.16975 0.019066
## 12 0.00315844     11   0.14313 0.16845 0.019422
## 13 0.00254194     12   0.13998 0.16476 0.019483
## 14 0.00179553     13   0.13743 0.16582 0.019552
## 15 0.00162380     14   0.13564 0.17413 0.021205
## 16 0.00148963     15   0.13401 0.17270 0.020963
## 17 0.00125920     19   0.12806 0.17259 0.021311
## 18 0.00125284     21   0.12554 0.17197 0.021234
## 19 0.00104439     22   0.12428 0.17172 0.021294
## 20 0.00092544     23   0.12324 0.17456 0.021821
## 21 0.00089991     24   0.12231 0.17534 0.021891
## 22 0.00089709     25   0.12141 0.17492 0.021892
## 23 0.00075949     26   0.12052 0.17567 0.022118
## 24 0.00063507     28   0.11900 0.17391 0.021954
## 25 0.00055323     29   0.11836 0.17308 0.021810
## 26 0.00047867     30   0.11781 0.17420 0.022452
## 27 0.00045871     31   0.11733 0.17465 0.022494
## 28 0.00043051     33   0.11641 0.17528 0.022518
## 29 0.00041062     34   0.11598 0.17642 0.022539
## 30 0.00040923     35   0.11557 0.17649 0.022539
## 31 0.00040008     36   0.11516 0.17649 0.022539
```

```
## 32 0.00037988     37   0.11476 0.17598 0.022537
## 33 0.00037974     38   0.11438 0.17592 0.022533
## 34 0.00037371     39   0.11400 0.17543 0.022469
## 35 0.00036741     40   0.11363 0.17525 0.022469
## 36 0.00035987     41   0.11326 0.17447 0.022461
## 37 0.00032487     43   0.11254 0.17477 0.022530
## 38 0.00031451     44   0.11222 0.17399 0.022465
## 39 0.00031431     45   0.11190 0.17409 0.022465
## 40 0.00031354     46   0.11159 0.17405 0.022466
## 41 0.00027113     47   0.11128 0.17397 0.022463
## 42 0.00021116     48   0.11100 0.17407 0.022486
## 43 0.00020894     49   0.11079 0.17370 0.022481
## 44 0.00019348     50   0.11058 0.17363 0.022481
## 45 0.00018863     51   0.11039 0.17395 0.022502
## 46 0.00018102     52   0.11020 0.17406 0.022502
## 47 0.00015307     53   0.11002 0.17414 0.022502
## 48 0.00014642     54   0.10987 0.17420 0.022501
## 49 0.00012987     55   0.10972 0.17442 0.022503
## 50 0.00012570     56   0.10959 0.17445 0.022501
## 51 0.00011911     57   0.10947 0.17408 0.022476
## 52 0.00011234     58   0.10935 0.17394 0.022477
## 53 0.00010176     59   0.10923 0.17407 0.022478
## 54 0.00010000     60   0.10913 0.17427 0.022484
```

```
m_tree <- prune(m_tree, cp = 0.002)
rpart.plot(m_tree)
```



Jak widać nasze drzewko treningowe różni się od drzewka z pełnego zbioru danych. Oczywiście wynika to z podziału zbioru gdzie w zbiorze *train* mam tylko 70% danych

3.1 Predykcja

Podstawiam zbiór testowy

```
pred <- predict(m_tree, newdata = test)
results <- data.frame(charges_log = log10(test$charges),pred_charges = pred)
View(results)

results <- data.frame(charges_log = log10(test$charges), pred_log = pred, charges = test$charges,
                      pred = 10^pred)
View(results)
```

MSE (mean squer error)Średni błąd kwadratowy: zmienna liczona dla zmiennych wykładniczych

```
mean((results$charges_log - results$pred_log)^2)
```

```
## [1] 0.03637355
```

RMSE(root mean squer error)błąd średnio kwadratowy dla zmiennych wykładniczych

```
sqrt(mean((results$charges_log - results$pred_log)^2))
```

```
## [1] 0.1907185
```

Można powiedzieć, że model myli się o 0,17log10(charges)

```
sqrt(mean((results$charges - results$pred)^2))
```

```
## [1] 5549.625
```

```
results <- data.frame(charges_log = log10(test$charges), pred_log = pred, charges = test$charges,
                      pred = 10^pred, diff = abs(test$charges - 10^pred))
head(results) %>%  round(3) %>%  kable
```

charges_log	pred_log	charges	pred	diff
4.227	4.246	16884.924	17601.344	716.420
3.237	3.322	1725.552	2096.956	371.403
4.342	3.647	21984.471	4431.577	17552.894
3.807	3.873	6406.411	7473.017	1066.606
4.461	4.117	28923.137	13100.627	15822.510
4.444	4.386	27808.725	24347.152	3461.574

Chociaż są wyniki znacznie odbiegające od oczekiwanych jak chociażby pierwszy z tabeli to

model srednio myli sie o ok. 4850$.