Final_TripAdvisor

David DeStephano

May 5, 2020

```
library(tidyverse)
library(caret)
library(colorspace)
library(circlize)
library(AppliedPredictiveModeling)
library(ModelMetrics)
library(rpart) #cart
library(rpart.plot)
```

Introduction

Gaugeing how a potential guest would rate your and other hotels could be a pertinent question within marketing. It could allow you to tailor experiences for a certain guest, suggest additional features/amenities for a higher or lower price that you know the guest would enjoy based on their tripadvisor history. In this exercise, the researchers intend to simply predict guest ratings of a hotel, but this exercise could be extended to a prescriptive analytic framework, in which guest experiences could be tailored based on their tripadvisor accounts.

This data was precleaned, but multilevel factors were dummified.

There are some issues with this data, weekdays are sometime miscoded as the month. When modeling, the number of rooms is perfectly 1 to 1 with the hotel name, so coefficients are NA automatically. Honestly not exactly sure why this is hosted on UCI's machine learning repository. Having months in the day of week variable is particularly bad.

Exploratory Analysis

Unfortunately there are few numeric variables about the hotel itself, so the numeric variables we do have show us more about the reviewers, namely, that reviewers that are very active on the site are less likely to leave negative reviews, or vice versa, that non-tripadvisor users are more likely to leave a negative review/people with no reviews will leave a bad review but not a good review.

Additionally, since scores can only be 1, 2, 3, 4, 5, scatter plots are not particularly useful for visualization.

Models

All predictor variables were included.

Random forest outpeformed the linear model appreciably, and marginally outperformed the decision tree model.

The results for both metrics adopted, MAE, can be seen in the file. In the scale from 1 to 5 used for the score on TripAdvisor, the random forest (the best model) achieved a MAE/average absolute deviation of 0.759. This tells us the model, on average, is within one star of predicting the real score.

```
trip<-read_csv("C:\\Users\\daved\\Documents\\Data Sceince II\\p8106_final_dd2948\\LasVegasTripAdvisorRe
```

Exploratory Analysis

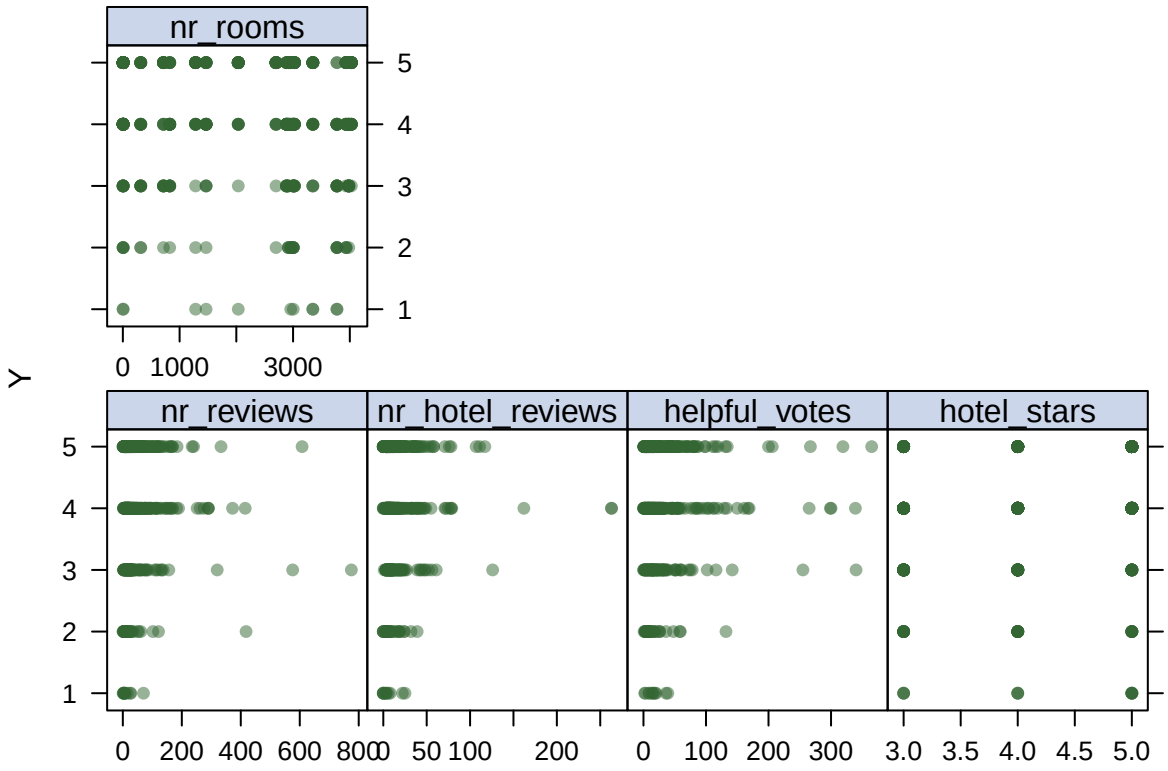
```
trip[sapply(trip, is.character)] <- lapply(trip[sapply(trip, is.character)],
                                       as.factor)


trip<-as.data.frame(trip)

trip2 <- trip[,-14]
```

Data viz here!!!

```
ggplot(trip, aes(x=traveler_type, y=score)) +
  geom_violin()
```


```
nums<-select_if(trip, is.numeric)
```

```
x <-model.matrix(score~.,nums)[,-1]

y <- nums$score

theme1 <-trellis.par.get()
theme1$plot.symbol$col <-rgb(.2, .4, .2, .5)
theme1$plot.symbol$pch <- 16
theme1$plot.line$col <-rgb(.8, .1, .1, 1)
theme1$plot.line$lwd <- 2
theme1$strip.background$col <-rgb(.0, .2, .6, .2)
trellis.par.set(theme1)

featurePlot(x, y, plot = "scatter", labels =c("","Y"),type =c("p"), layout =c(4, 2))
```

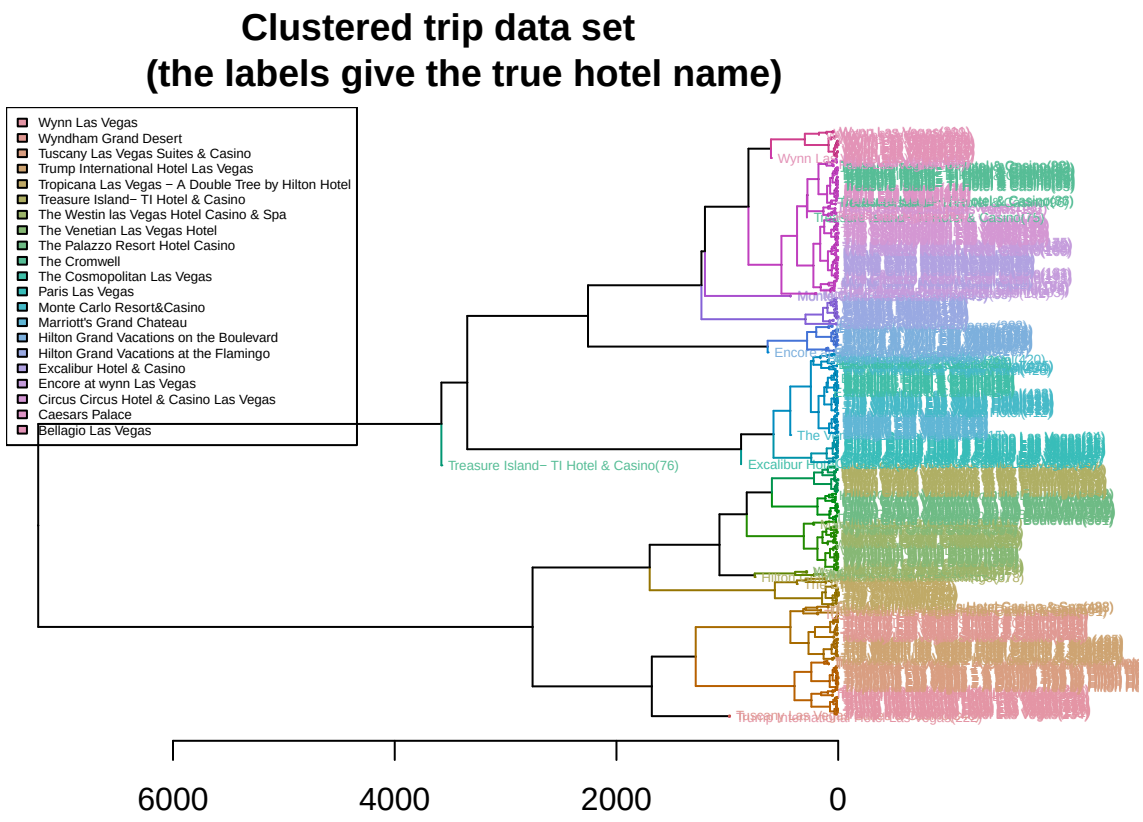


Unfortunately there are few numeric variables about the hotel itself, so the numeric variables we do have show us more about the reviewers, namely, that reviewers that are very active on the site are less likely to leave negative reviews, or vice versa, that non-tripadvisor users are more likely to leave a negative review/people with no reviews will leave a bad review if they had a particularly bad experience.

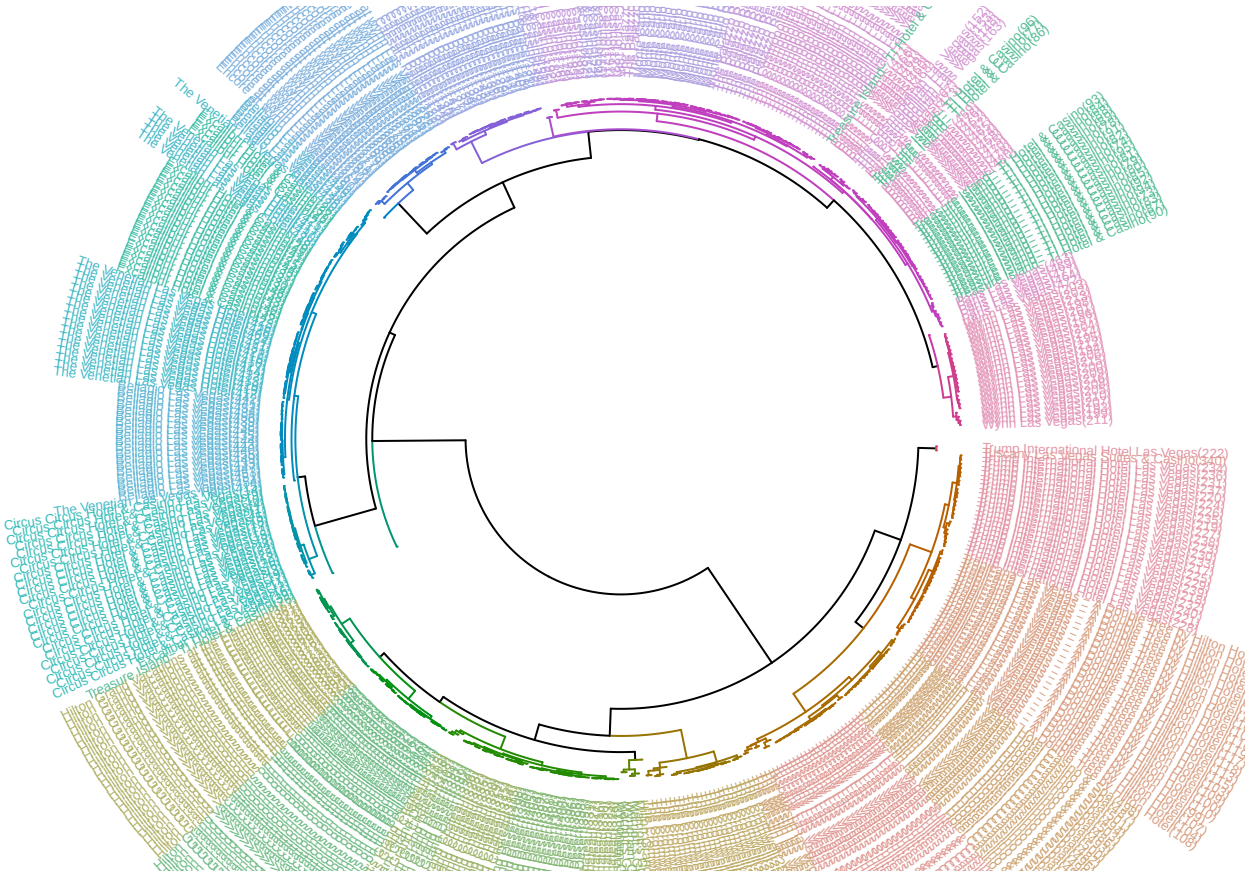
First let's try hierarchical clustering to see how hotels cluster together

```
d_trip <- dist(trip2) # method="man" # is a bit better
hc_trip <- hclust(d_trip, method = "complete")
trip_hotel <- rev(levels(trip[,14]))
```

```r
library(dendextend)
dend <- as.dendrogram(hc_trip)

# Color the branches based on the clusters:
dend <- color_branches(dend, k=21) #, groupLabels=trip_hotel)

labels_colors(dend) <-
   rainbow_hcl(21)[sort_levels_values(
      as.numeric(trip[,14])[order.dendrogram(dend)]
   )]

# add the hotel name to the labels:
labels(dend) <- paste(as.character(trip[,14])[order.dendrogram(dend)],
                           "(",labels(dend),")",
                           sep = "")
dend <- hang.dendrogram(dend,hang_height=0.1)
# reduce the size of the labels:
#dend <- assign_values_to_leaves_nodePar(dend, 0.05, "lab.cex")
dend <- set(dend, "labels_cex", 0.4)
# And plot:
par(mar = c(3,3,3,7))
plot(dend,
     main = "Clustered trip data set
     (the labels give the true hotel name)",
     horiz =  TRUE,  nodePar = list(cex = .007))
legend("topleft", legend = trip_hotel, fill = rainbow_hcl(21), cex=0.4)
```




```
par(mar = rep(0,4))
circlize_dendrogram(dend)
```



```
#hc.complete <- hclust(dist(trip), method = "complete")
#hc.average <- hclust(dist(trip), method = "average")
#hc.single <- hclust(dist(trip), method = "single")
#hc.centroid <- hclust(dist(trip), method = "centroid")
```

Models

dummify the variables

```
dmy<- dummyVars(" ~.", data=trip, fullRank = T)
trsf <- data.frame(predict(dmy, newdata = trip))
```

Create test and train datasets

```
set.seed(1)
rowTrain = createDataPartition(trsf$score,
                               p=2/3,
                               list=F)

train <- trsf[rowTrain, ]
test <- trsf[-rowTrain, ]
```

Linear regression

```
ctrl1 <-trainControl(method = "cv", number = 5)
set.seed(1)

lm.fit <-train(score~.,
                data = train,
                method = "lm",
                trControl = ctrl1)



getTrainPerf(lm.fit)
```

```
##   TrainRMSE TrainRsquared  TrainMAE method
## 1  1.306348    0.05292255 0.9922389     lm
```

```
predy.lm <-predict(lm.fit, newdata = test)



mae(test$score, predy.lm)
```

```
## [1] 0.9931578
```

```
mse(test$score, predy.lm)
```

```
## [1] 1.544906
```

```
rmse(test$score, predy.lm)
```

```
## [1] 1.242942
```

```
coef(lm.fit$finalModel) %>% knitr::kable()
```

	x
(Intercept)	-3.0304515
user_country.Belgium	-1.2229839
user_country.Brazil	0.4274518
user_country.Canada	0.0342448
user_country.China	NA
user_country.Costa.Rica	-0.2878910
user_country.Croatia	-0.2619384
user_country.Czech.Republic	NA
user_country.Denmark	NA
user_country.Egypt	0.4680765
user_country.Finland	1.1225213
user_country.France	NA
user_country.Germany	-0.4136332

	x
user_country.Greece	0.2707227
user_country.Hawaii	-0.2658649
user_country.Honduras	0.2113779
user_country.Hungary	NA
user_country.India	0.8509233
user_country.Iran	-0.0655927
user_country.Ireland	0.8225911
user_country.Israel	-0.3662229
user_country.Italy	0.7996831
user_country.Japan	NA
user_country.Jordan	NA
user_country.Kenya	0.9503907
user_country.Korea	0.5604872
user_country.Kuwait	-0.4684576
user_country.Malaysia	0.2215665
user_country.Mexico	0.8329425
user_country.Netherlands	0.2410564
user_country.New.Zeland	-0.3756631
user_country.Norway	2.3303631
user_country.Phillippines	-0.4256321
user_country.Puerto.Rico	1.7682251
user_country.Saudi.Arabia	-0.3902348
user_country.Scotland	NA
user_country.Singapore	0.6572433
user_country.South.Africa	1.4061156
user_country.Spain	NA
user_country.Swiss	NA
user_country.Switzerland	-0.6704874
user_country.Syria	1.2305782
user_country.Taiwan	NA
user_country.Thailand	0.3789970
user_country.UK	0.7607791
user_country.United.Arab.Emirates	0.5731039
user_country.USA	0.1237400
nr_reviews	-0.0002962
nr_hotel_reviews	-0.0020750
helpful_votes	0.0011712
period_of_stay.Jun.Aug	1.7659064
period_of_stay.Mar.May	0.9750093
period_of_stay.Sep.Nov	1.7150955
traveler_type.Couples	0.6211953
traveler_type.Families	0.3077215
traveler_type.Friends	0.7283454
traveler_type.Solo	0.1665940
pool.YES	3.7739100
gym.YES	0.4218418
tennis_court.YES	0.4960251
spa.YES	-2.4249510
casino.YES	0.4348146
free_internet.YES	1.0653864
hotel_name.Caesars.Palace	-0.2081853
hotel_name.Circus.Circus.Hotel. . . Casino.Las.Vegas	NA

	x
hotel_name.Encore.at.wynn.Las.Vegas	0.2338325
hotel_name.Excalibur.Hotel. . . Casino	-0.5051049
hotel_name.Hilton.Grand.Vacations.at.the.Flamingo	-2.0458809
hotel_name.Hilton.Grand.Vacations.on.the.Boulevard	1.9175103
hotel_name.Marriott.s.Grand.Chateau	NA
hotel_name.Monte.Carlo.Resort.Casino	NA
hotel_name.Paris.Las.Vegas	-0.3367670
hotel_name.The.Cosmopolitan.Las.Vegas	0.0381435
hotel_name.The.Cromwell	NA
hotel_name.The.Palazzo.Resort.Hotel.Casino	0.2769801
hotel_name.The.Venetian.Las.Vegas.Hotel	0.4147647
hotel_name.The.Westin.las.Vegas.Hotel.Casino. . . Spa	-0.1458258
hotel_name.Treasure.Island..TI.Hotel. . . Casino	-0.7066554
hotel_name.Tropicana.Las.Vegas. . . A.Double.Tree.by.Hilton.Hotel	-0.1073797
hotel_name.Trump.International.Hotel.Las.Vegas	0.6106982
hotel_name.Tuscany.Las.Vegas.Suites. . . Casino	-0.3555871
hotel_name.Wyndham.Grand.Desert	NA
hotel_name.Wynn.Las.Vegas	NA
hotel_stars	NA
nr_rooms	NA
user_continent.188	NA
user_continent.732	NA
user_continent.787	NA
user_continent.Africa	NA
user_continent.Asia	0.2245523
user_continent.Europe	0.5254540
user_continent.North.America	1.0991382
user_continent.Oceania	0.9023702
user_continent.South.America	NA
member_years.0	0.1576860
member_years.1	0.3022066
member_years.10	0.3809522
member_years.11	0.3269515
member_years.12	1.5697110
member_years.13	0.1953702
member_years.2	0.7410319
member_years.3	0.3925781
member_years.4	0.2520241
member_years.5	0.2378683
member_years.6	0.4989149
member_years.7	0.0768568
member_years.8	0.0016745
member_years.9	NA
member_years.Asia	NA
member_years.Europe	NA
member_years.North.America	-0.2460532
member_years.Oceania	NA
member_years.South.America	NA
review_month.1	0.7446055
review_month.10	0.5949382
review_month.11	2.2036027
review_month.2	0.1137449

	x
review_month.3	0.6916194
review_month.4	0.3025934
review_month.5	0.8186898
review_month.6	0.8229566
review_month.7	0.5284106
review_month.8	0.5726766
review_month.9	1.9858074
review_month.April	0.5569618
review_month.August	0.1467734
review_month.December	1.6988534
review_month.February	1.7574192
review_month.January	2.0140050
review_month.July	-0.1275406
review_month.June	-0.0514520
review_month.March	1.4515608
review_month.May	0.8604811
review_month.November	0.1446356
review_month.October	0.1407214
review_month.September	NA
review_weekday.August	-0.7431911
review_weekday.December	1.1057445
review_weekday.February	1.7651734
review_weekday.Friday	0.3178565
review_weekday.January	0.3923335
review_weekday.July	-1.7305519
review_weekday.June	NA
review_weekday.March	-0.6229304
review_weekday.May	-0.8232691
review_weekday.Monday	0.1333102
review_weekday.November	-1.2397148
review_weekday.October	-2.8058416
review_weekday.Saturday	0.6513155
review_weekday.September	NA
review_weekday.Sunday	0.2086641
review_weekday.Thursday	0.6450504
review_weekday.Tuesday	-0.3165103
review_weekday.Wednesday	NA

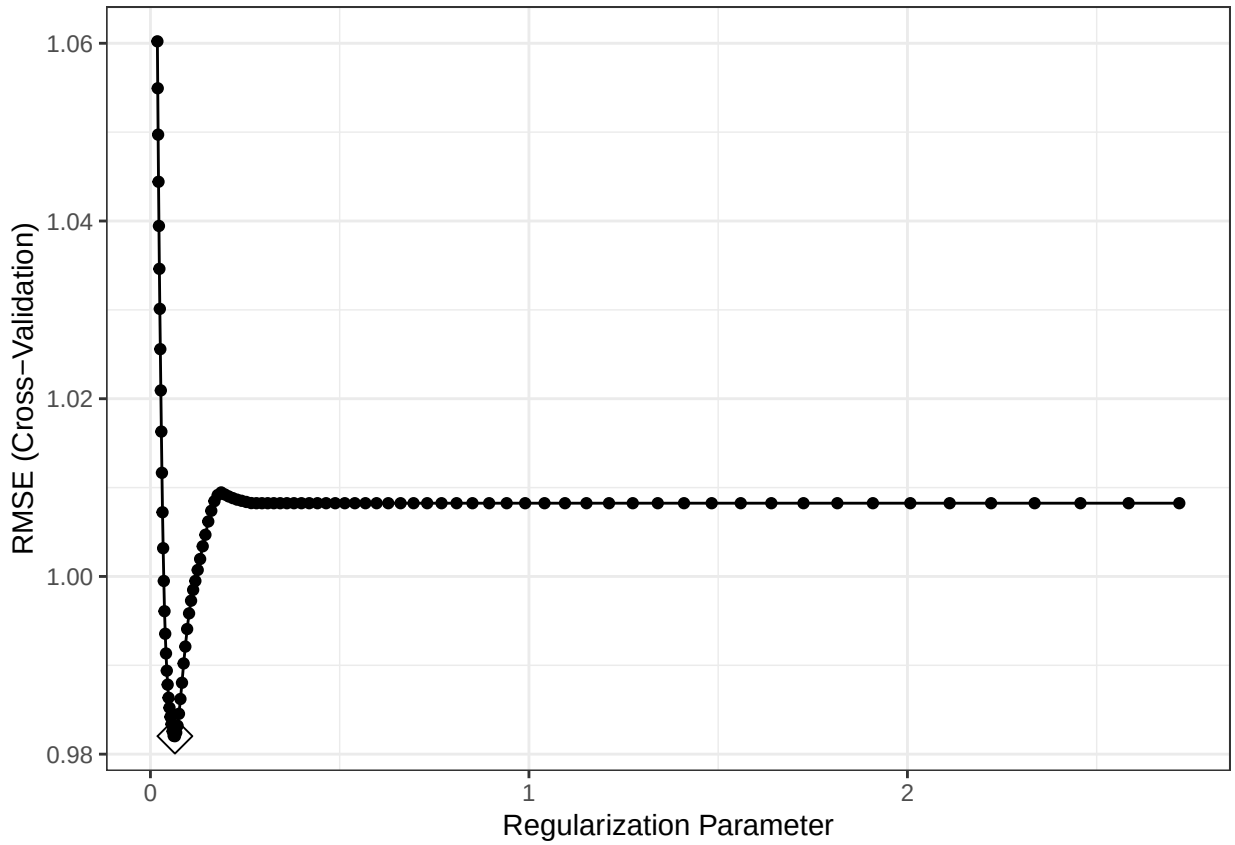
Lasso Regression

```
set.seed(1)
lasso.fit <- train(score~.,
                   data=train,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 1,
                                          lambda = exp(seq(-4, 1, length=100))),
                   preProc = c("center", "scale"),
trControl = ctrl1)
```

```
## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
```

```
## uniqueCut = 10, : These variables have zero variances: user_country.China,
## user_country.Croatia, user_country.Czech.Republic, user_country.Denmark,
## user_country.France, user_country.Hungary, user_country.Japan,
## user_country.Jordan, user_country.Korea, user_country.Kuwait,
## user_country.Scotland, user_country.Spain, user_country.Swiss,
## user_country.Taiwan

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19, uniqueCut
## = 10, : These variables have zero variances: user_country.Belgium,
## user_country.China, user_country.Czech.Republic, user_country.Denmark,
## user_country.Egypt, user_country.Finland, user_country.France,
## user_country.Hungary, user_country.Japan, user_country.Jordan,
## user_country.Kenya, user_country.Phillippines, user_country.Scotland,
## user_country.Spain, user_country.Swiss, user_country.Switzerland,
## user_country.Taiwan, member_years.12, member_years.South.America

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: user_country.China,
## user_country.Czech.Republic, user_country.Denmark, user_country.France,
## user_country.Honduras, user_country.Hungary, user_country.Japan,
## user_country.Jordan, user_country.Scotland, user_country.Spain,
## user_country.Swiss, user_country.Syria, user_country.Taiwan,
## user_country.United.Arab.Emirates, member_years.13

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: user_country.China,
## user_country.Czech.Republic, user_country.Denmark, user_country.France,
## user_country.Greece, user_country.Hungary, user_country.Iran,
## user_country.Italy, user_country.Japan, user_country.Jordan,
## user_country.Puerto.Rico, user_country.Scotland, user_country.South.Africa,
## user_country.Spain, user_country.Swiss, user_country.Taiwan, review_month.
## 11

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: user_country.China,
## user_country.Czech.Republic, user_country.Denmark, user_country.France,
## user_country.Hungary, user_country.Japan, user_country.Jordan,
## user_country.Saudi.Arabia, user_country.Scotland, user_country.Spain,
## user_country.Swiss, user_country.Taiwan

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
## trainInfo, : There were missing values in resampled performance measures.

## Warning in preProcess.default(thresh = 0.95, k = 5, freqCut = 19,
## uniqueCut = 10, : These variables have zero variances: user_country.China,
## user_country.Czech.Republic, user_country.Denmark, user_country.France,
## user_country.Hungary, user_country.Japan, user_country.Jordan,
## user_country.Scotland, user_country.Spain, user_country.Swiss,
## user_country.Taiwan
```

```
ggplot(lasso.fit, highlight = TRUE) + theme_bw()
```




```
lasso.fit$bestTune
```

```
##    alpha     lambda
## 26     1 0.06474015
```

```
predy.lasso <-predict(lasso.fit, newdata = test)

mae(test$score, predy.lasso)
```

```
## [1] 0.7682081
```

```
mse(test$score, predy.lasso)
```

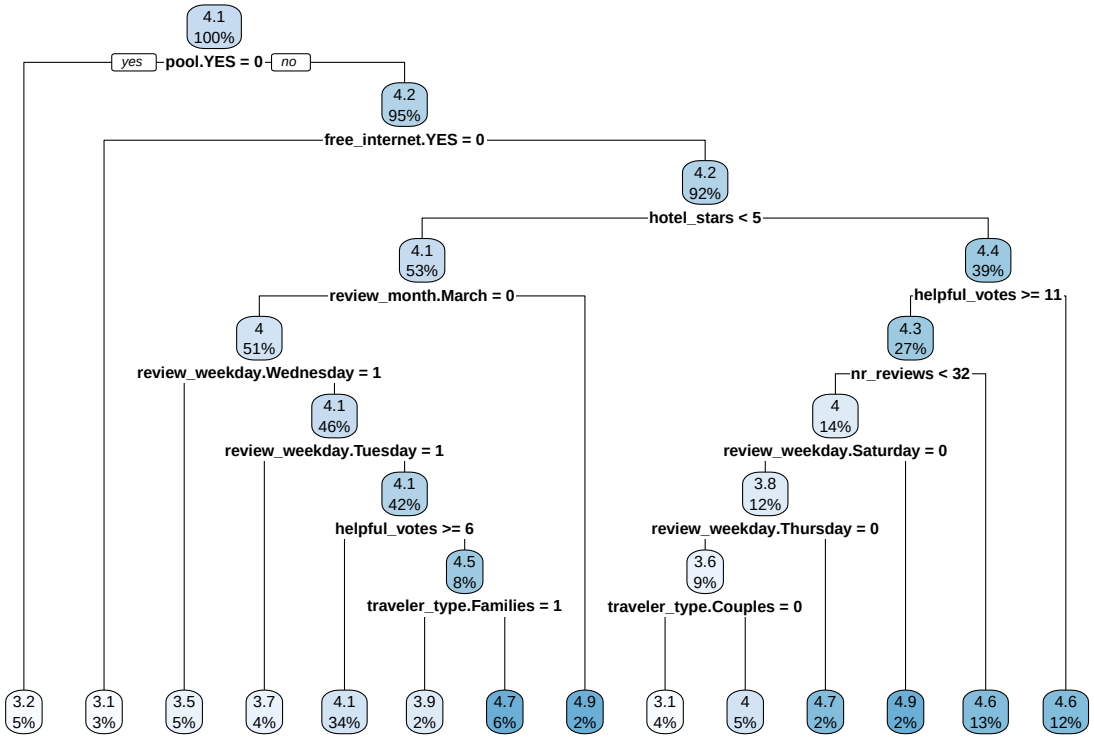
```
## [1] 0.9316627
```

```
rmse(test$score, predy.lasso)
```

```
## [1] 0.9652267
```

Decision Tree

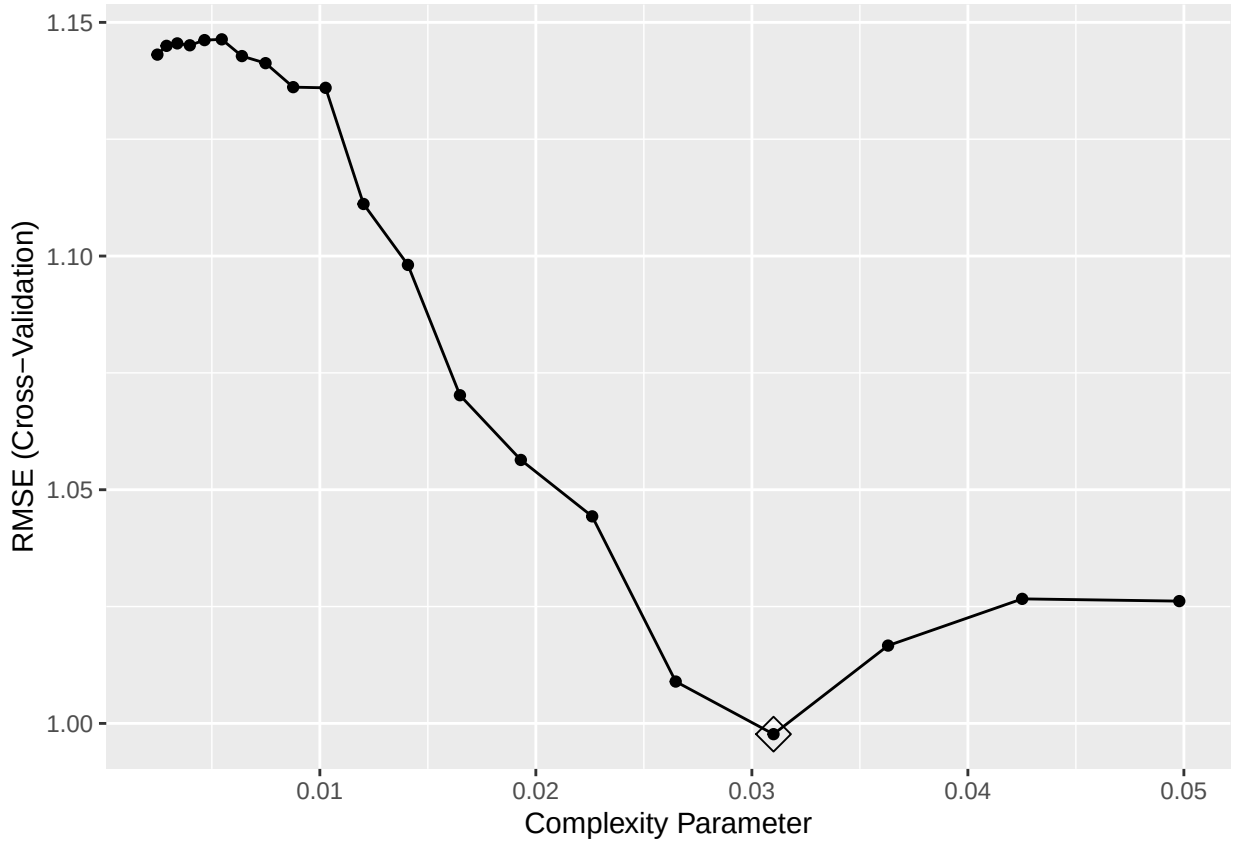
```
set.seed(1)
tree1 <- rpart(formula = score~., data = train)
rpart.plot(tree1)
```




```
set.seed(1)
rpart.fit <-train(score~., train,
                  method = "rpart",
                  tuneGrid =data.frame(cp =exp(seq(-6,-3, length = 20))),
                  trControl = ctrl1)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
## trainInfo, : There were missing values in resampled performance measures.
```

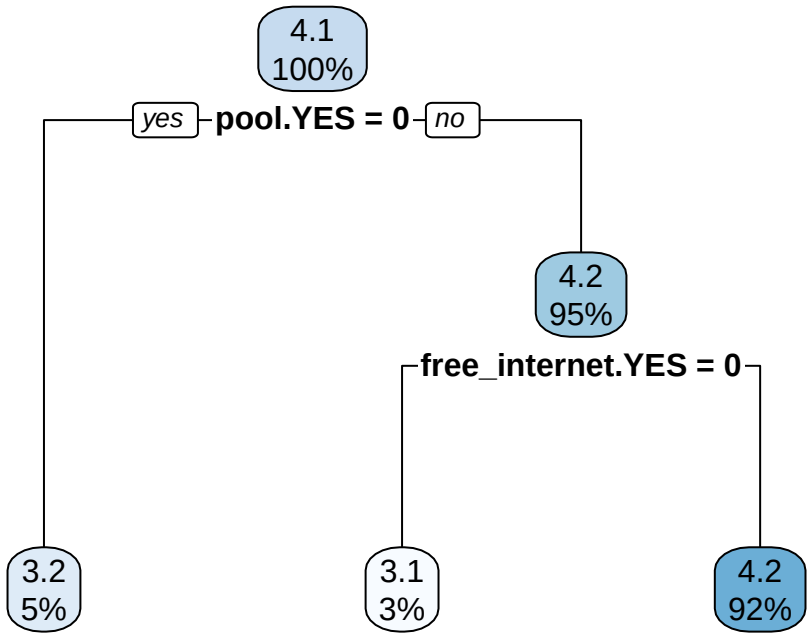
```
ggplot(rpart.fit, highlight =TRUE)
```




```
rpart.fit$bestTune
```

```
##           cp
## 17 0.0310026
```

```
rpart.plot(rpart.fit$finalModel)
```

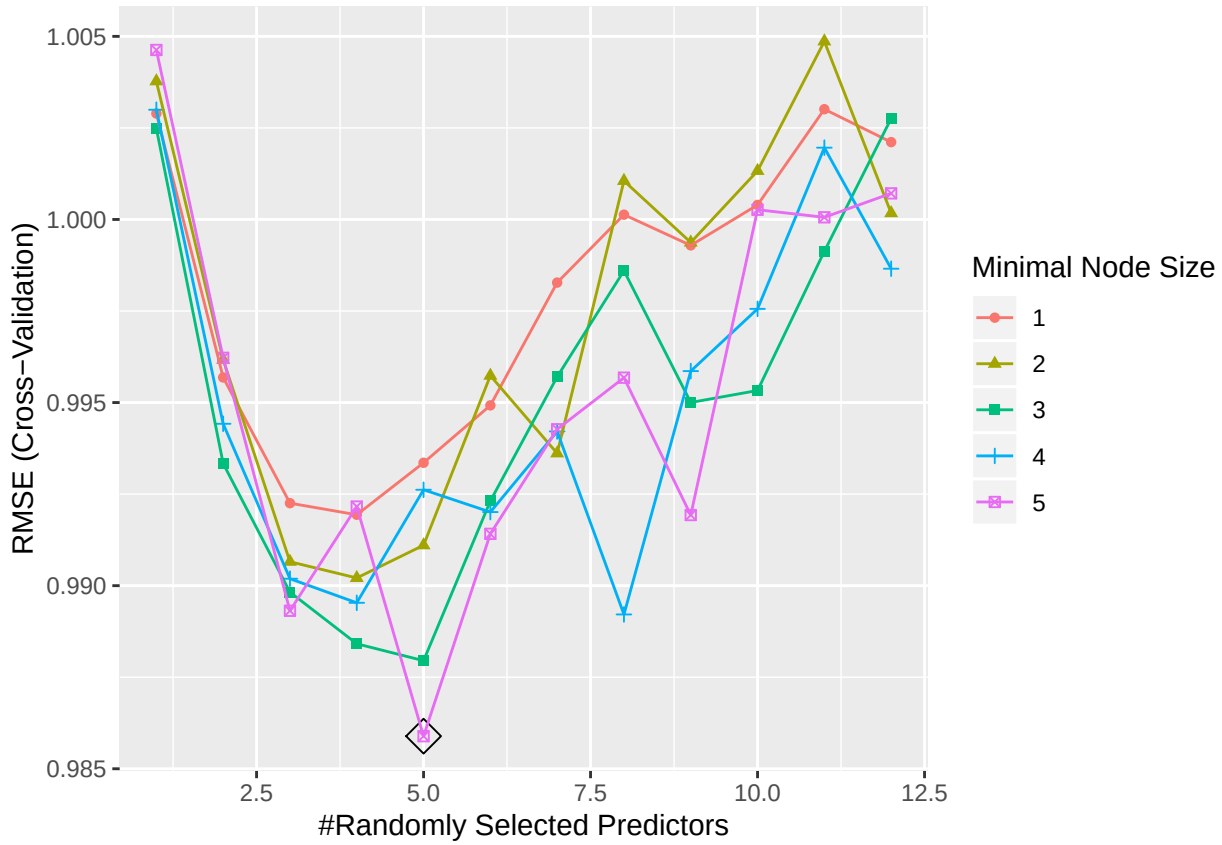



```
predy2.rpart <-predict(rpart.fit, newdata = test)
mse(predy2.rpart, test$score)
```

```
## [1] 0.9265384
```

RF

```
rf.grid <-expand.grid(mtry = 1:12,
                      splitrule = "variance",
                      min.node.size = 1:5)

set.seed(1)

rf.fit <-train(score~., train,
               method = "ranger",
               tuneGrid=rf.grid,
               trControl=ctrl1,
               importance="permutation")

ggplot(rf.fit, highlight = TRUE)
```




```
rfImp <- varImp(rf.fit, scale = FALSE)
rfImp
```

```
## ranger variable importance
##
##   only 20 most important variables shown (out of 153)
##
##                                                Overall
## hotel_stars                                   0.021292
## helpful_votes                                 0.020448
## pool.YES                                      0.016666
## hotel_name.Monte.Carlo.Resort.Casino          0.015857
## nr_reviews                                    0.014236
## hotel_name.Circus.Circus.Hotel...Casino.Las.Vegas 0.013492
## free_internet.YES                             0.010584
## review_weekday.Thursday                       0.005226
## review_weekday.June                           0.004544
## review_weekday.February                       0.004222
## spa.YES                                       0.003932
## hotel_name.The.Venetian.Las.Vegas.Hotel       0.003336
## user_continent.Asia                           0.002901
## nr_hotel_reviews                              0.002658
## review_weekday.Saturday                       0.002444
## hotel_name.Treasure.Island..TI.Hotel...Casino 0.002310
## hotel_name.Excalibur.Hotel...Casino           0.002235
```

```
## review_month.March                                0.002190
## period_of_stay.Mar.May                            0.001879
## traveler_type.Couples                             0.001857
```

```
predy.rf <-predict(rf.fit, newdata = test)
```

```
mae(test$score, predy.rf)
```

```
## [1] 0.7505002
```

```
mse(test$score, predy.rf)
```

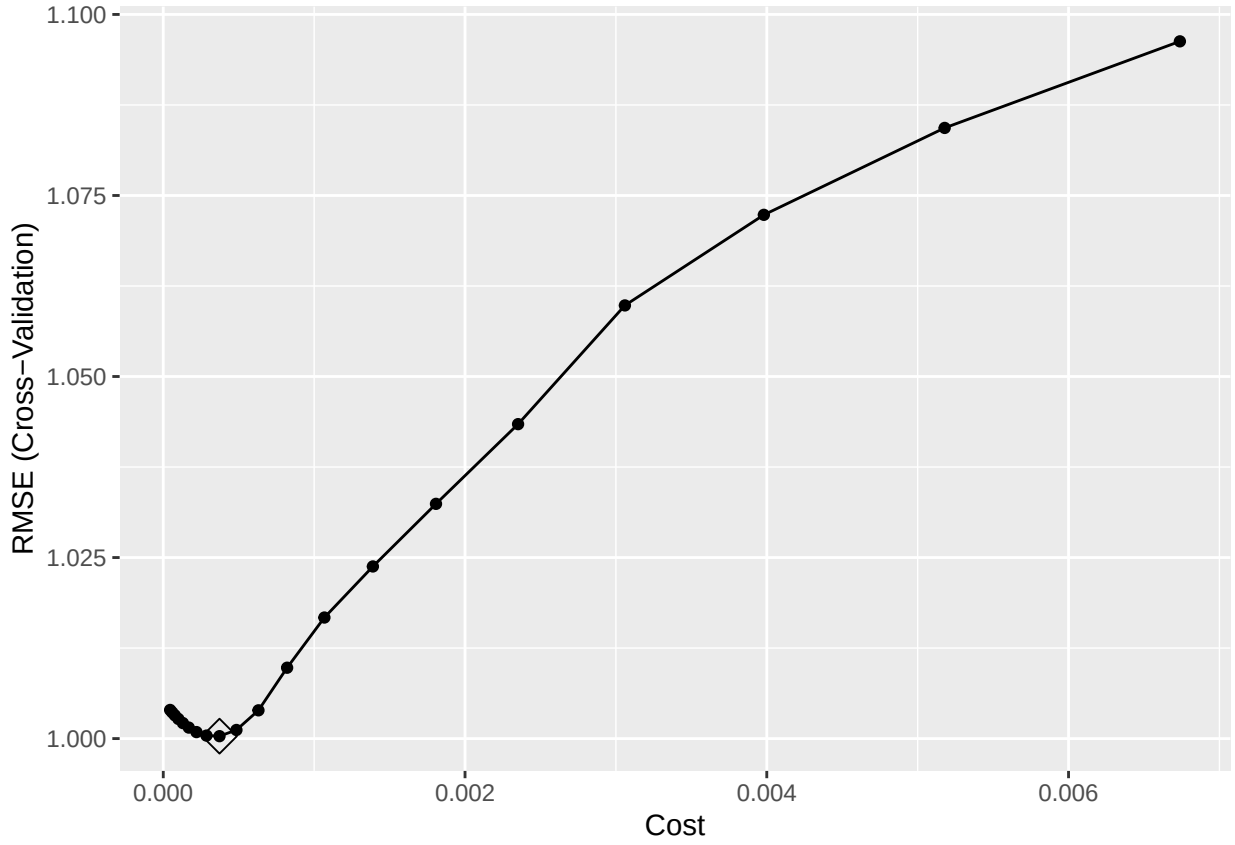
```
## [1] 0.8746935
```

```
rmse(test$score, predy.rf)
```

```
## [1] 0.9352505
```

SVM/Support Vector Regression

```
svr.fit <- train(score ~ .,  data = train,
                 method= "svmLinear2",
                 preProcess = c("center", "scale"),
                 tuneGrid = data.frame(cost = exp(seq(-10,-5,len=20))),
                 trControl = ctrl1)

ggplot(svr.fit, highlight = TRUE)
```




```
predy.svr <-predict(svr.fit, newdata = test)



mae(test$score, predy.svr)
```

```
## [1] 0.7678199
```

```
mse(test$score, predy.svr)
```

```
## [1] 0.9630293
```

```
rmse(test$score, predy.svr)
```

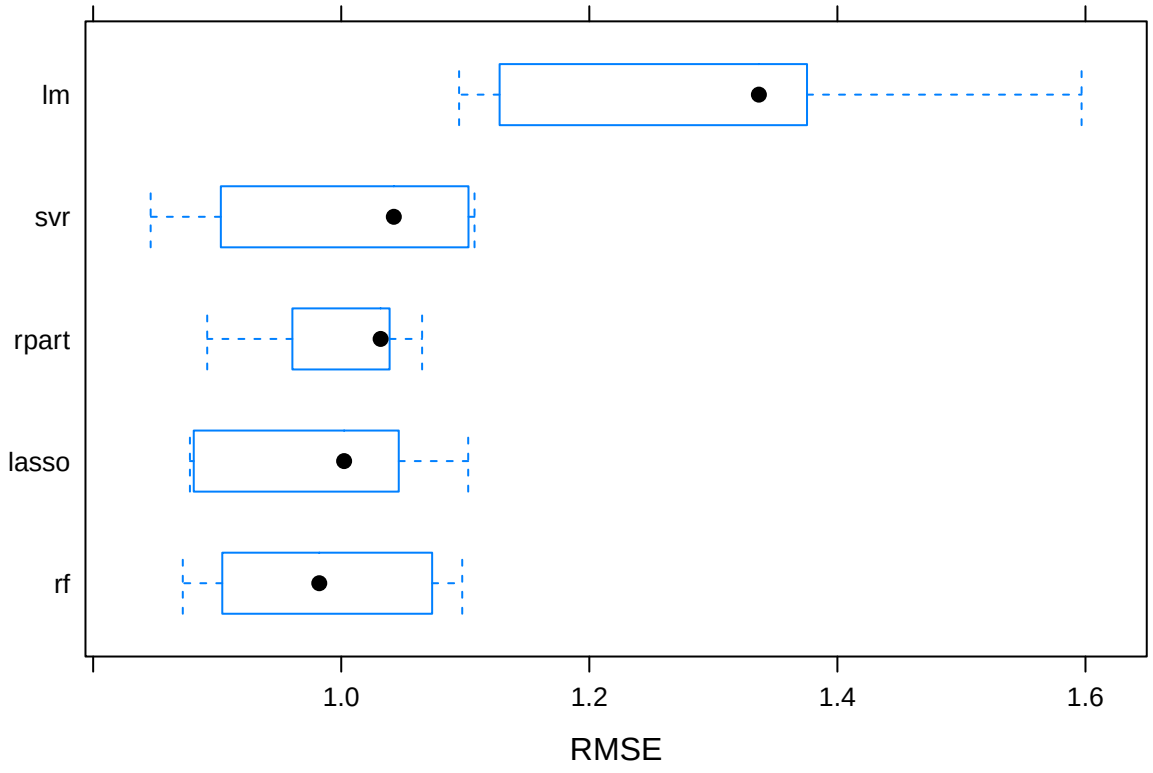
```
## [1] 0.9813406
```

Compare models

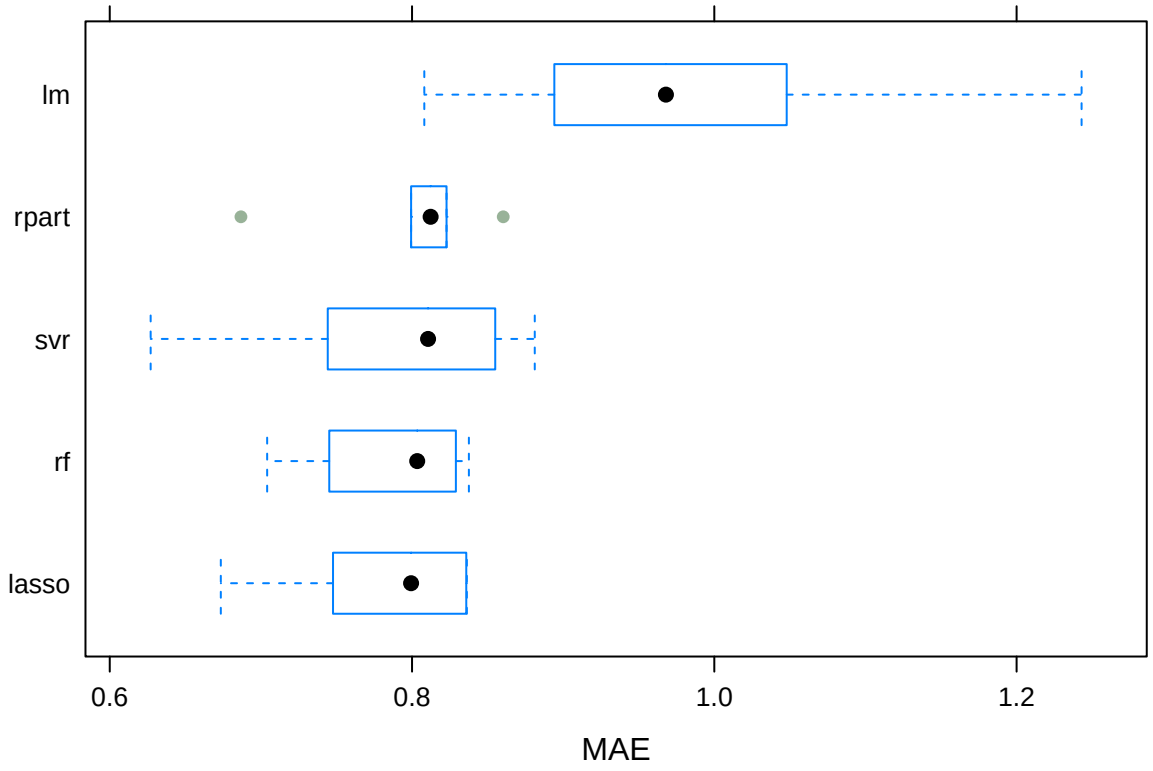
```
resamp <-resamples(list(lm = lm.fit, lasso=lasso.fit,rpart = rpart.fit, rf=rf.fit, svr=svr.fit))
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: lm, lasso, rpart, rf, svr
## Number of resamples: 5
##
## MAE
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lm    0.8081305 0.8941618 0.9679954 0.9922389 1.0478958 1.2430108    0
## lasso 0.6734899 0.7477364 0.7993729 0.7785502 0.8358878 0.8362639    0
## rpart 0.6868089 0.7993910 0.8123122 0.7963393 0.8228348 0.8603495    0
## rf    0.7041404 0.7452358 0.8034313 0.7838882 0.8290122 0.8376215    0
## svr   0.6270562 0.7442334 0.8106077 0.7836315 0.8550538 0.8812065    0
##
## RMSE
##            Min.   1st Qu.    Median      Mean  3rd Qu.     Max. NA's
## lm    1.0950068 1.1276649 1.3366973 1.3063480 1.375510 1.596861    0
## lasso 0.8780174 0.8810881 1.0023242 0.9820264 1.046360 1.102342    0
## rpart 0.8919470 0.9606035 1.0317417 0.9976985 1.038996 1.065204    0
## rf    0.8722504 0.9040974 0.9822762 0.9858903 1.073288 1.097539    0
## svr   0.8463373 0.9029196 1.0423925 1.0003331 1.102583 1.107433    0
##
## Rsquared
##                Min.      1st Qu.      Median       Mean    3rd Qu.
## lm    1.665863e-04 0.0142399181 0.068478581 0.05292255 0.07654282
## lasso 3.320931e-04 0.0009199601 0.101614921 0.10540911 0.13612528
## rpart 5.305271e-06 0.0097350086 0.023148103 0.07557629 0.12792302
## rf    1.075577e-04 0.0036447348 0.078013403 0.09060697 0.14512162
## svr   6.128837e-05 0.0020506945 0.009853243 0.02277111 0.04496643
##             Max. NA's
## lm    0.10518485    0
## lasso 0.28805330    0
## rpart 0.21707001    0
## rf    0.22614753    0
## svr   0.05692388    0
```

```
bwplot((resamp), metric = "RMSE")
```




```
bwplot((resamp), metric = "MAE")
```


lm
rpart
svr
rf
lasso
0.6
0.8
1.0
1.2
MAE