Housing Analysis Tutorial

Brad Rafferty

5/17/2020

We are going to analyze a curated dataset of various variables regarding housing in California.

```
## Require gbm
library(gbm)

## Loaded gbm 2.1.5

library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

set.seed(20200509)

## Read in the data
dir_data = './data/'
col_names = c('Median House Value', 'Median Income', 'Housing Median Age',
'Average Num Rooms', 'Average Num Bedrooms', 'Population', 'Average
Occupancy', 'Latitude', 'Longitude')
data_in = read.csv(file = paste0(dir_data, 'housing_data.csv'), col.names =
col_names)
```

Looking at the original raw data, some occupancy numbers go over 100! One even goes over 1200! This could be data entry error, or the inclusion of high-rise apartment complexes, for example. Such cases are not meant for this dataset, and so they are excluded using the a filter.

```
data_in <- data_in %>%
  dplyr::filter(Average.Occupancy < 12)
```

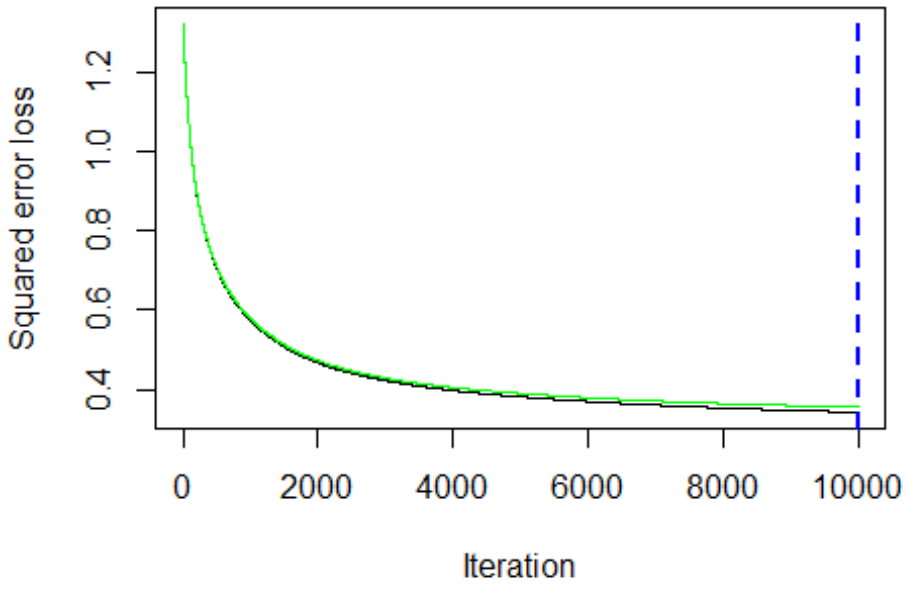
Now we divide the data into a training set and a test set.

```
## Set up training samples
num_obs <- nrow(data_in)
```

```
data_in <- data_in[sample(num_obs), ]
train_size <- floor(0.75*num_obs)
set.seed(20200510)
idx_train <- sample(seq_len(num_obs), size = train_size) # indices of train
observations

## Split into train and test datasets
x_train <- data_in[idx_train, ] # train dataset
x_test <- data_in[-idx_train, ] # test dataset
```

With the data prepared, we model the training data with a GBM.

```
## Delcare parameters for gbm model
num_trees = 10000
shrinkage = 0.005
cv_folds  = 5
distr     = 'gaussian'

## Build model on train data using gbm with cross-validation
mdl <- gbm(Median.House.Value ~., data = x_train, n.tree = num_trees,
shrinkage = shrinkage, cv.folds = cv_folds, distribution = distr, verbose =
FALSE)
best_iter <- gbm.perf(object = mdl) # Pull the best iteration of the model
```



Using our model, we can analyze the prediction accuracy against the training data and the test data.

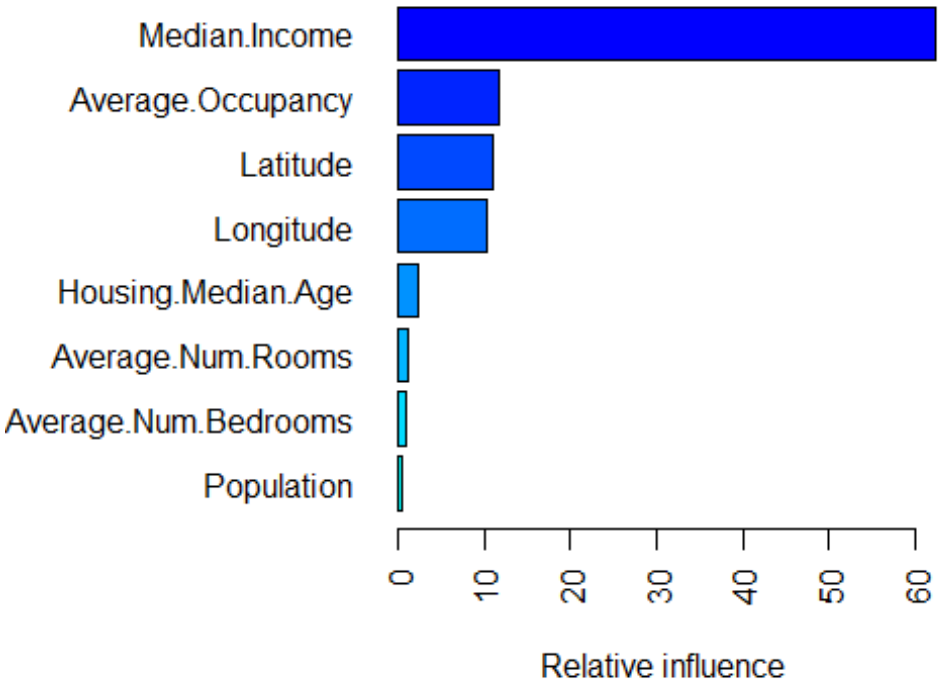
```r
# Predict on the new data using the "best" number of trees; by default,
# predictions will be on the link scale
median_house_value_train_hat <- predict(mdl, newdata = x_train, n.trees =
best_iter, type = "link")
median_house_value_test_hat <- predict(mdl, newdata = x_test, n.trees =
best_iter, type = "link")

## Mean squared error loss (prediction accuracy)
msg_pred_acc_train <- sprintf('The mean squared error of the gbm prediction
on the training dataset is $%0.0f', sum((x_train$Median.House.Value -
median_house_value_train_hat)^2)/nrow(x_train)*100000) # MSE, multiply by
100000 to get back into units of $ instead of $100k
print(msg_pred_acc_train)
```

```
## [1] "The mean squared error of the gbm prediction on the training dataset
is $34182"
```

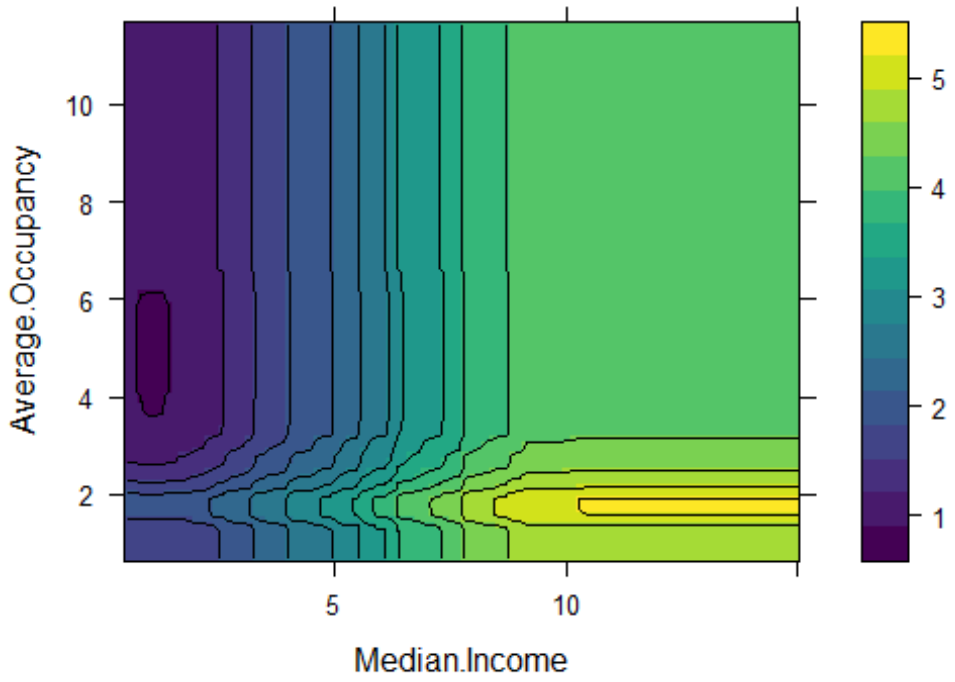
```r
msg_pred_acc_test <- sprintf('The mean squared error of the gbm prediction on
the test dataset is $%0.0f', sum((x_test$Median.House.Value -
median_house_value_test_hat)^2)/nrow(x_test)*100000)
print(msg_pred_acc_test)
```

```
## [1] "The mean squared error of the gbm prediction on the test dataset is
$33893"
```

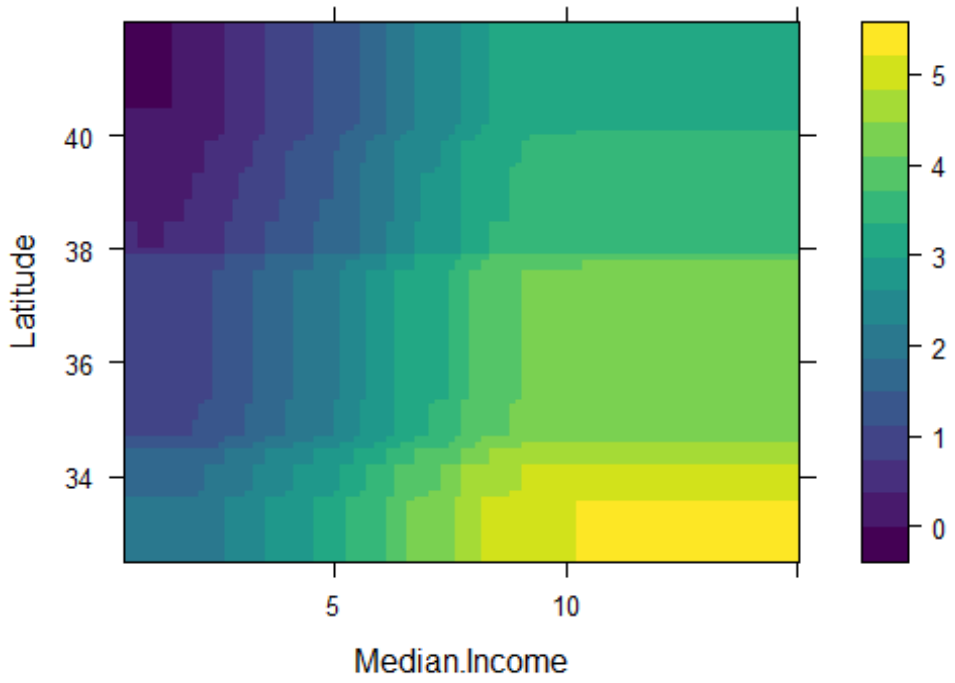
```r
## Relative influence of predictors
par(mar = c(5,10,1,1))
summary(mdl,
        cBars = ncol(x_train)-1,
        method = relative.influence,
        las = 2)
```



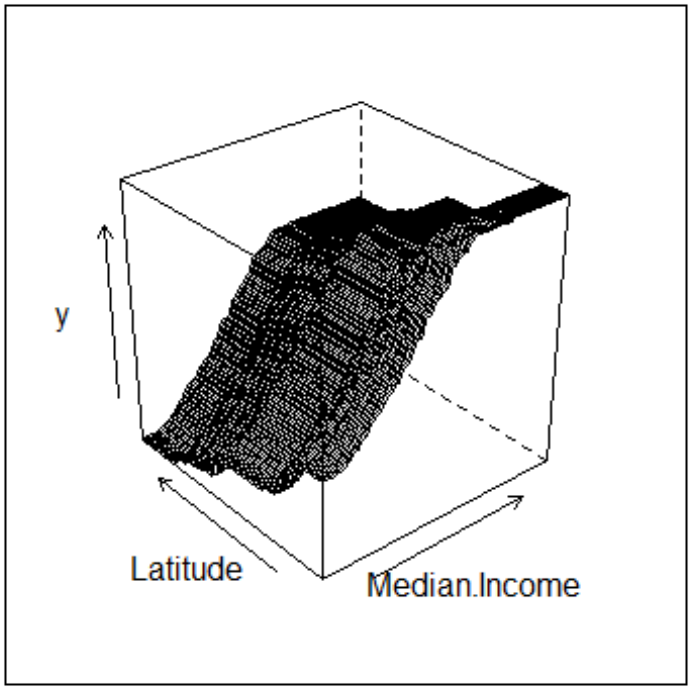

```
##                                       var    rel.inf
## Median.Income               Median.Income 62.3969515
## Average.Occupancy       Average.Occupancy 11.7787038
## Latitude                         Latitude 10.8785344
## Longitude                       Longitude 10.3062564
## Housing.Median.Age     Housing.Median.Age  2.2590960
## Average.Num.Rooms       Average.Num.Rooms  1.0807881
## Average.Num.Bedrooms Average.Num.Bedrooms  0.8497204
## Population                     Population  0.4499494

## Partial Dependences
plot(mdl, i.var = c(1,6), level.plot = TRUE, contour = TRUE)
```



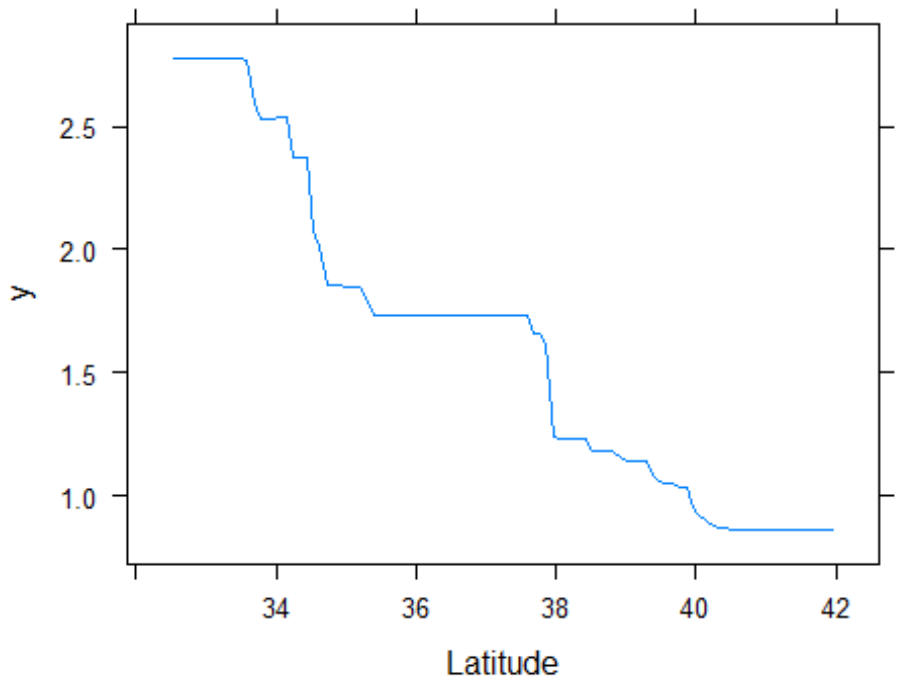

```
plot(mdl, i.var = c(1,7), level.plot = TRUE)
```



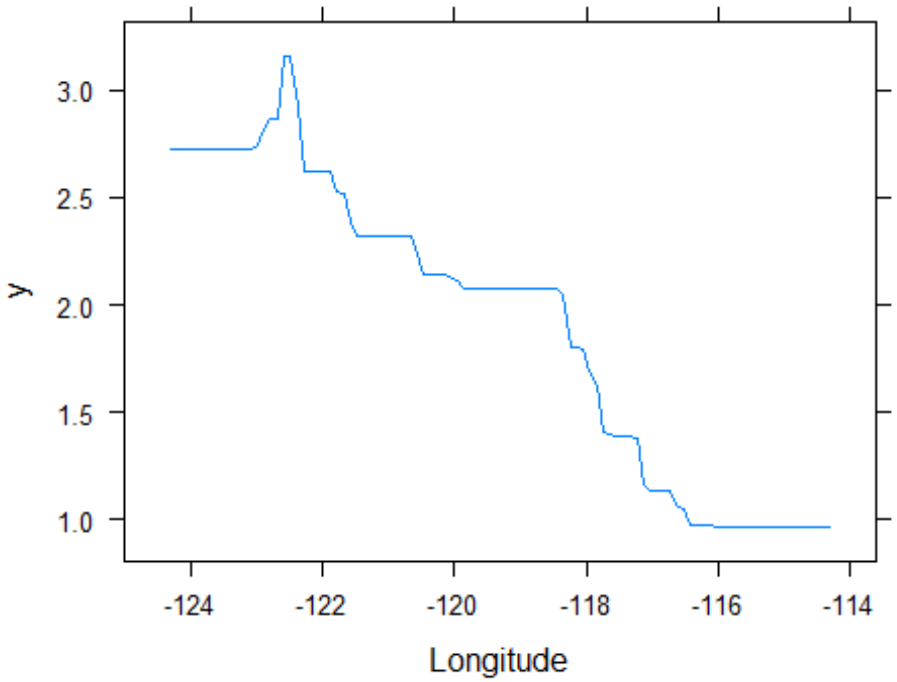

```
plot(mdl, i.var = c(1,7), level.plot = FALSE)
```




```
plot(mdl, i.var = 7)
```




```
plot(mdl, i.var = 8)
```



According to the relative influence measures, the most important variables in predicting the median house value are *Median Income*, *Average Occupancy*, and location (*Latitude* and *longitude*). Very surprisingly, the *Number of Bedrooms* in the house is relatively non-influential. One may expect that *Number of Bedrooms* would correlate with size of the house, which would be a large influence on the prediction. Apparently not!

Based on the marginal effects plot of *Median Income* with *Average Occupancy*, it appears that the occupancy has little influence on the *Median House Value* for a given median income.

On the other hand, *Latitude* (location) does have a large influence on the *Median House Value* for a given *Median Income*. This is further supported by the partial dependnece plot of only *Latitude* and only *Longitude*.