Health Insurance Machine Learning Analysis

2024-03-07

Machine Learning Analysis on individuals' health insurance charges in the US. The data includes both categorical and numeric measures. I will be providing a thorough regression analysis attempting to predict the 'charges' variable using the remainder of the predictors in the data set. Which model is most likely to provide the lowest MSE in the long-run? Which model would I choose if I was consulting with an insurance company on this data set?

```
insurance <- read.csv("insurance.csv", stringsAsFactors=TRUE)
head(insurance)
```

```
##   age    sex    bmi children smoker    region   charges
## 1  19 female 27.900        0    yes southwest 16884.924
## 2  18   male 33.770        1     no southeast  1725.552
## 3  28   male 33.000        3     no southeast  4449.462
## 4  33   male 22.705        0     no northwest 21984.471
## 5  32   male 28.880        0     no northwest  3866.855
## 6  31 female 25.740        0     no southeast  3756.622
```

Linear Model:

```
is.index <- sample(1:nrow(insurance), 0.7 * nrow(insurance))
is.train <- insurance[is.index, ]
is.test <- insurance[-is.index, ]
```

```
train.insurance.lm <- lm(charges~., data = is.train)
test.insurance.lm <- lm(charges~., data = is.test)

train.predictions <- predict(train.insurance.lm, newdata = is.train)

train.residuals <- is.train$charges - train.predictions

train.mse <- mean(train.residuals^2)

test.predictions <- predict(test.insurance.lm, newdata = is.test)

test.residuals <- is.test$charges - test.predictions

test.mse <- mean(test.residuals^2)

# Display MSE for training and test sets
cat("Training MSE:", train.mse, "\n")
```

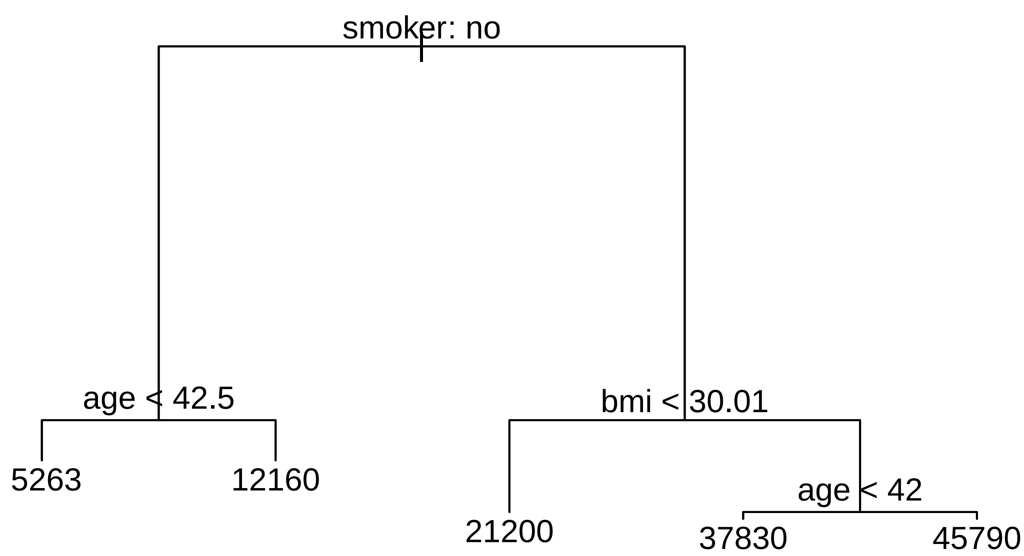
```
## Training MSE: 33593394
```

```r
cat("Test MSE:", test.mse, "\n")
```

```
## Test MSE: 41643538
```

Trees:

```r
library(tree)
insurance_tree <- tree(charges~., data=is.train)
plot(insurance_tree)
text(insurance_tree, pretty=0)
```



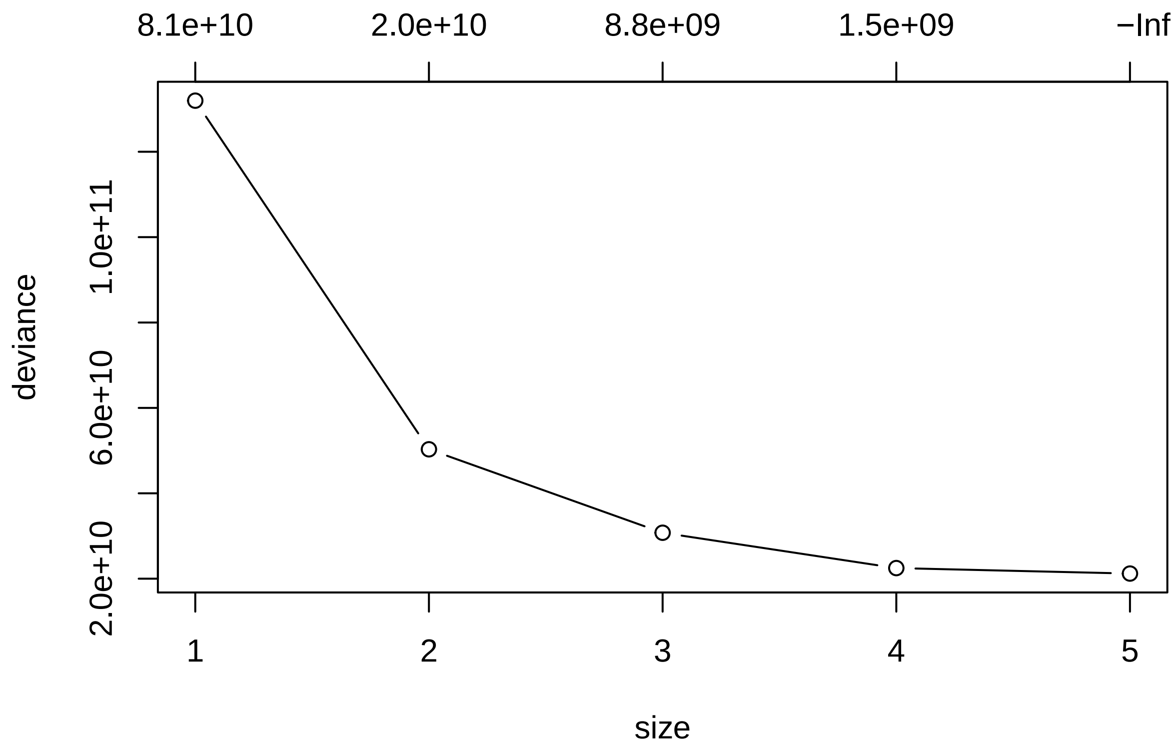

```r
summary(insurance_tree)
```

```
##
## Regression tree:
## tree(formula = charges ~ ., data = is.train)
## Variables actually used in tree construction:
## [1] "smoker" "age"    "bmi"
## Number of terminal nodes:  5
```

```
## Residual mean deviance:  21340000 = 1.987e+10 / 931
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -8368.0 -2889.0  -898.7     0.0  1094.0 24750.0
```

Residual mean deviance = 19830000

```r
cv.insurance_tree <- cv.tree(insurance_tree, K = 10)
plot(cv.insurance_tree, type="b")
```



Pruning is not necessary

```r
training_tree_MSE <- min(cv.insurance_tree$dev)/nrow(insurance)
cat("The Training MSE is", training_tree_MSE)
```

```
## The Training MSE is 15845338
```

```r
test_predict <- predict(insurance_tree, newdata = is.test)
test_MSE <- mean((is.test$charges - test_predict)^2)
cat( "The Testing MSE is", test_MSE)
```

```
## The Testing MSE is 30180017
```

Random Forest

```
library("randomForest")
```

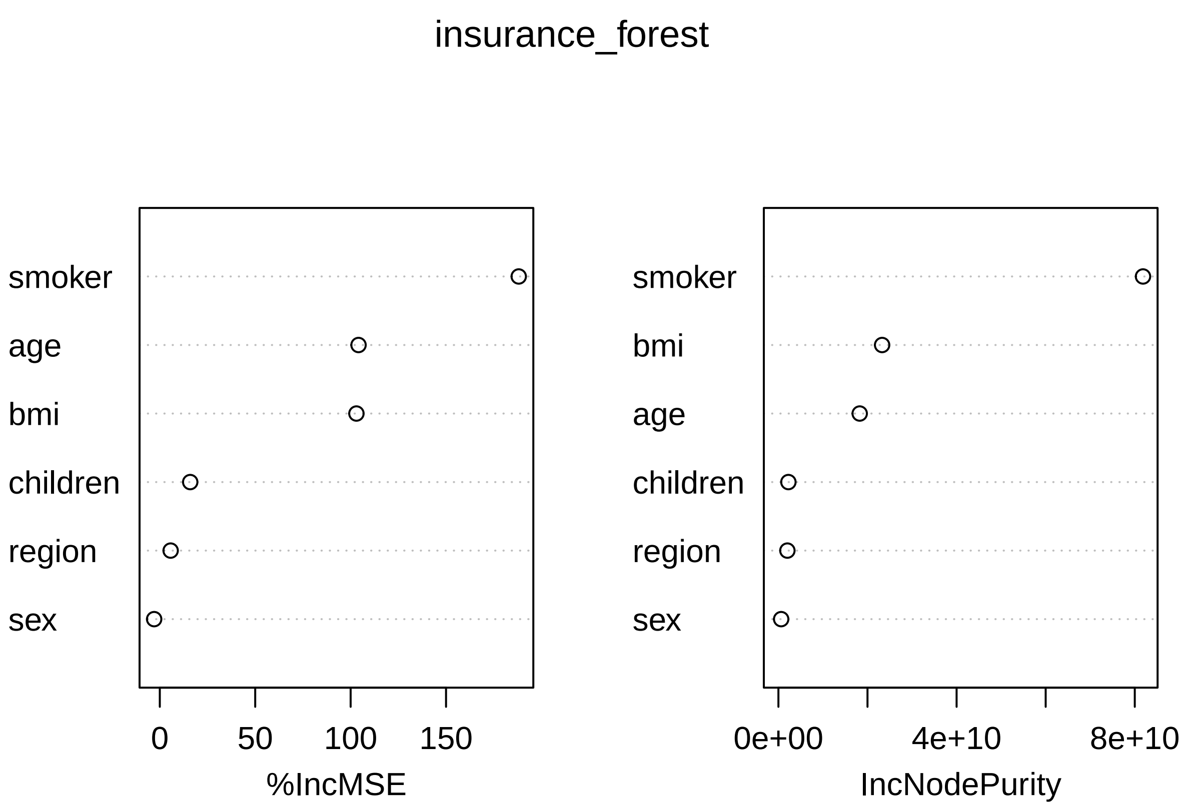
```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
insurance_forest <- randomForest(charges~., data=is.train, mtry=4, importance=TRUE)
print(insurance_forest)
```

```
##
## Call:
##  randomForest(formula = charges ~ ., data = is.train, mtry = 4,      importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##           Mean of squared residuals: 19925863
##                     % Var explained: 85.83
```

```
varImpPlot(insurance_forest)
```



```r
insurance_forest_train_MSE <- insurance_forest$mse[500]
cat("The training MSE is", insurance_forest_train_MSE)
```

```
## The training MSE is 19925863
```

```r
insurance_forest_predict <- predict(insurance_forest, newdata = is.test)
insurance_test_MSE <- mean((is.test$charges - insurance_forest_predict)^2)
cat("The testing MSE is", insurance_test_MSE)
```

```
## The testing MSE is 26613167
```

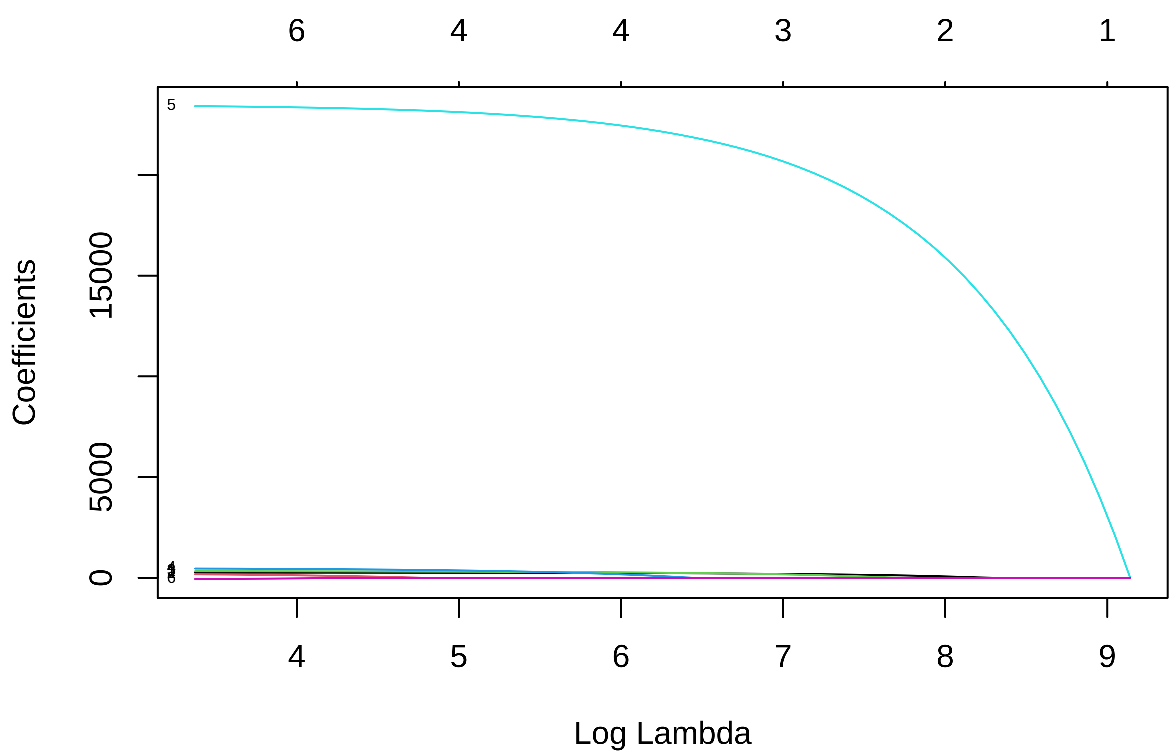
Lasso

```r
library(glmnet)
```

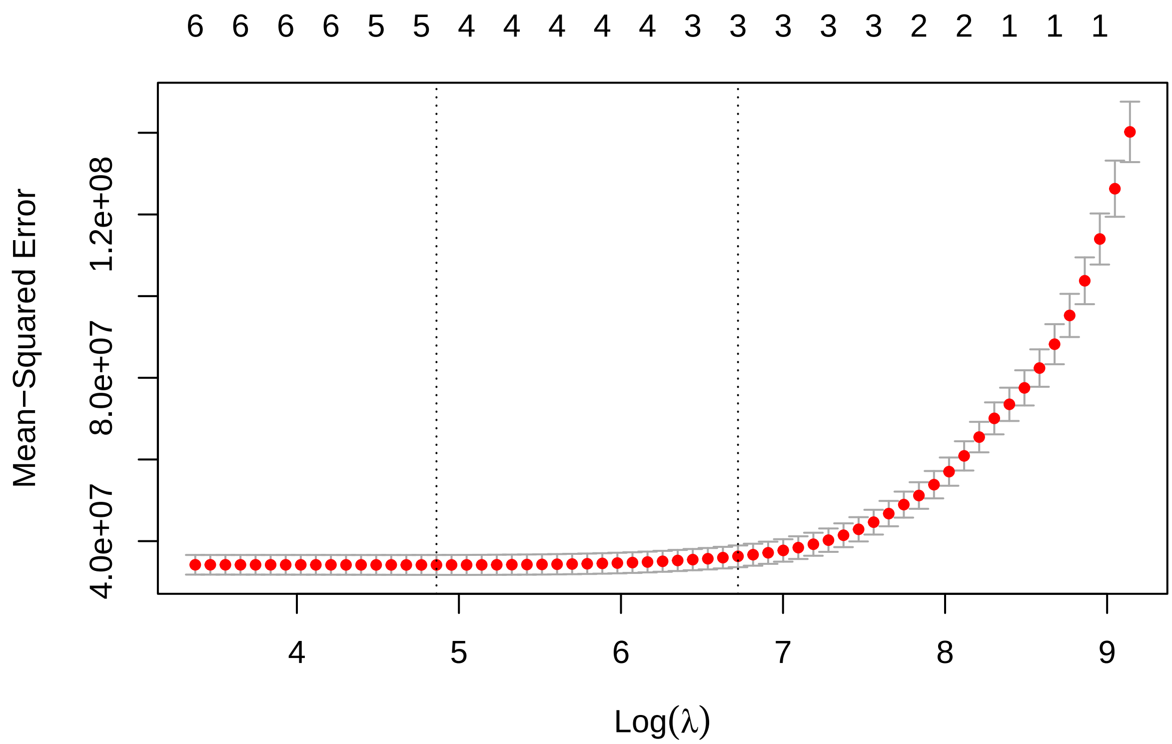
```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```r
x <- data.matrix(is.train[,c('age', 'sex', 'bmi', 'children', 'smoker', 'region')])
y <- is.train$charges
insurance_lasso <- cv.glmnet(x, y, alpha=1)
plot(insurance_lasso$glmnet.fit, label=TRUE, xvar="lambda")
```



```r
plot(insurance_lasso)
```




```r
lambda_value <- insurance_lasso$lambda.min
lambda_value
```

```
## [1] 129.2132
```

```r
library(glmnet)

lasso_prediction <- predict(insurance_lasso, s="lambda.min", newx=data.matrix(is.test[,c('age', 'sex',

lasso_MSE <- mean((lasso_prediction - is.test$charges)^2)
lasso_MSE
```

```
## [1] 44109566
```

Boosting

```r
library(gbm)
```

```
## Loaded gbm 2.1.8.1
```

```r
insurance_boosting <- gbm(charges ~ ., data = is.train, distribution = "gaussian", n.trees = 5000, cv.fo
insurance_boosting
```

```
## gbm(formula = charges ~ ., distribution = "gaussian", data = is.train,
##     n.trees = 5000, interaction.depth = 2, shrinkage = 0.1, cv.folds = 10)
## A gradient boosted model with gaussian loss function.
## 5000 iterations were performed.
## The best cross-validation iteration was 100.
## There were 6 predictors of which 6 had non-zero influence.
```

```r
insurance_boosting_predictions <- predict(insurance_boosting, newdata = is.test)
```

```
## Using 100 trees...
```

```r
boosting_MSE <- mean((is.test$charges - insurance_boosting_predictions)^2)
cat("The testing MSE is", boosting_MSE)
```

```
## The testing MSE is 26112544
```

Explanation:

The lowest testing MSE is given by the Boosting model. This is the best model and it will provide the lowest MSE in the long run. The decision tree is the best model to choose if consulting with an insurance company because its the most simple one and the easiest to explain however its not the most reliable model.