Statistical Analysis of Medical Costs with Machine Learning

Tayyab Ali

Introduction

Health insurance companies form a giant industry that affects every person with health insurance. According to S&P Global Market Intelligence, the life insurance sector had net premiums totaling $600.6 billion in 2018. To make their profits, insurance companies charge a higher premium than the amount payed to the insured person. These companies spend much a lot of time and money to predict health care costs. A patient's cost for treatment can depend on many types of factors: location, type fo clinic, diagnosis, type of treatment, bmi, etc.

The data I am using for my project is about medical costs and includes the amount charged. There are a lot of factors that go into predicting a person's medical charges, and I hope to find which variables are important in this prediction. I am interested in the factors can help explain the medical costs for patients. My outcome variable is going to be "charges" and I will use the six remaining variables as my predictive covariates. The goal of this project is to find the relation between "charges" and multiple predicitive covariates by using machine learning methods.

For my exploratory analysis, I visualized the different variables through histograms, boxplots, and scatterplots. After learning which variables were important in predicting medical charges, I created five linear regression models and analyzed their R-squared values to find the most accurate one. I also used shrinkage methods and analyzed their performance by comparing their Mean Squared Error (MSE).

Training and testing data were created to use other machine learning models. I used Bagging, Random Forest, Boost, and XGboost models. Importance matrix were generated from these models that emphsized certain variables were more important in predicting charges than others. I also created Predicted Charge vs. Actual charge plots to see how well the models could predict medical charges.

Below is a list of the packages I used for this project.

```
library(ISLR)
library(dplyr)
library(ggplot2)
library(glmnet)
library(dplyr)
library(tree)
library(randomForest)
library(doMC)
library(gbm)
library(xgboost)
```

Summary Statistics

Charges- Individual medical costs billed by health insurance

Age- age of primary beneficiary

Sex- insurance contractor gender, female, male

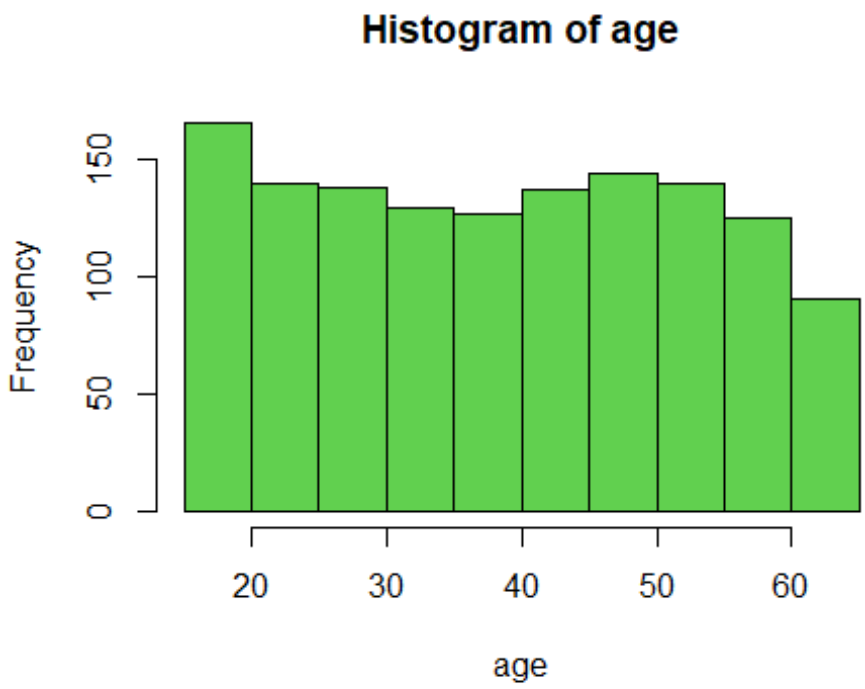
BMI- Body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9

Children- Number of children covered by health insurance / Number of dependents

Smoker- Smoking

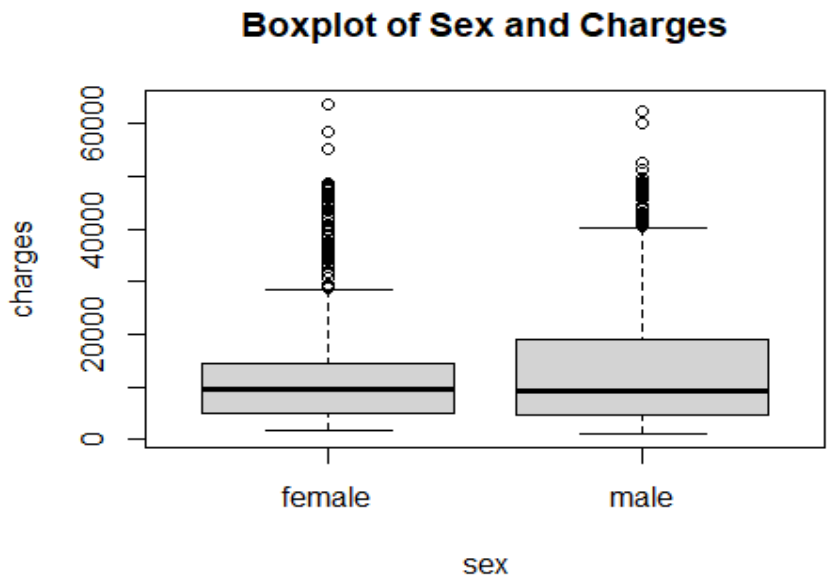
Region- the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.

```
insurance <- read.csv("insurance.csv")
attach(insurance)

hist(age, col = 3, breaks = 15 )
```



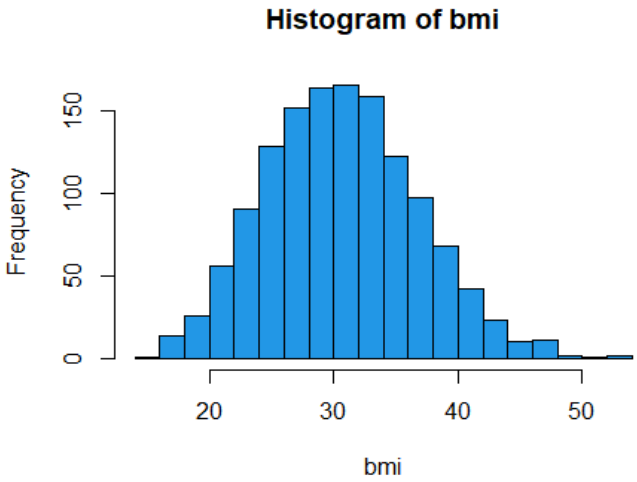
The histrogram of age shows the age distribution in my data set. We have a pretty even distribution of age groups, however, there are slightly more people in the 0-20 age range than the 60+ age range.

```
sex = as.factor(sex)
plot(sex, charges, main = "Boxplot of Sex and Charges", xlab = "sex", ylab =
"charges")
```



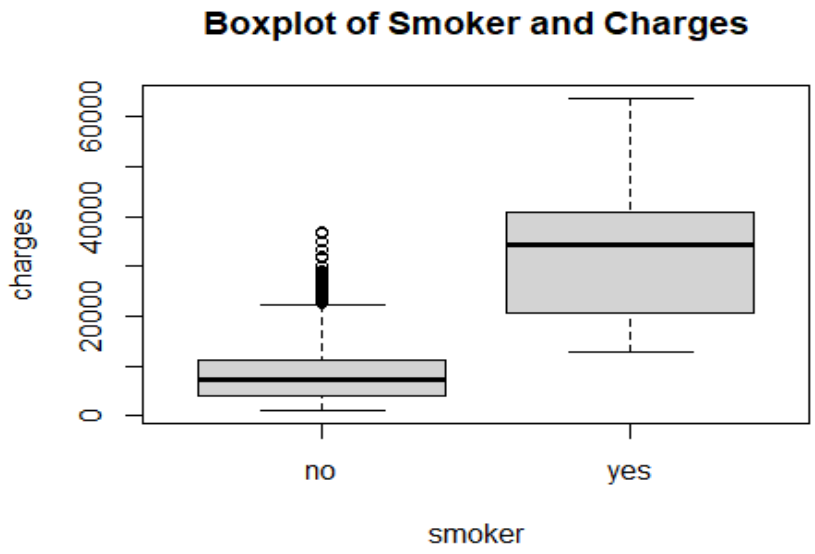
The boxplot of sex vs charges shows that median amount of charges for males and females is almost identical. Looking at the Quartile 3 values, the boxplot of the sex variables might indicate that males pay slightly more in charges than females.

```
hist(bmi,col = 4, breaks = 20)
```



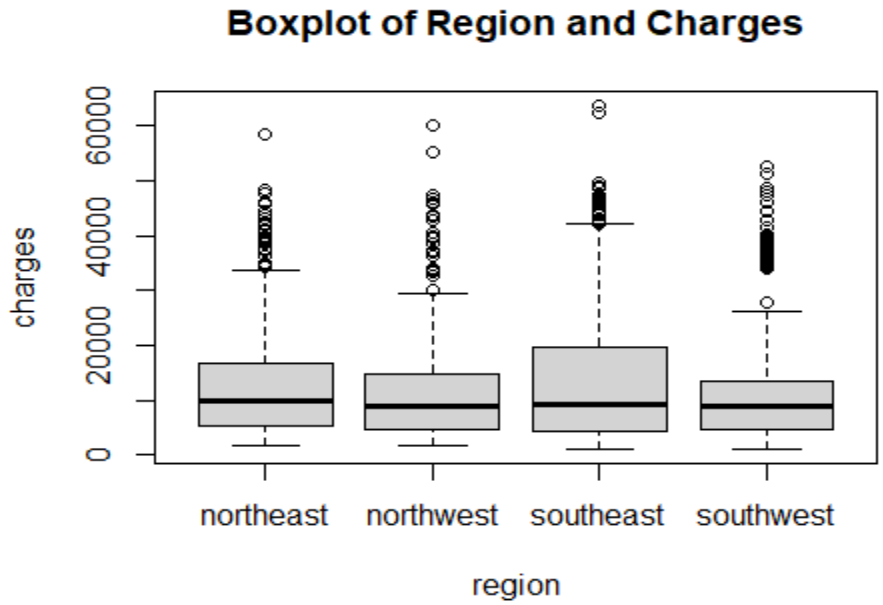
The histogram of BMI shows that the Body Mass Index ration in normally distributed around a mean of 30. This value is actually above the ideal BMI values which ranges from 18.5 to 24.9. This means that most of our sample is not in the ideal BMI range.

```
smoker = as.factor(smoker)
plot(smoker, charges, main = "Boxplot of Smoker and Charges", xlab =
"smoker", ylab = "charges")
```



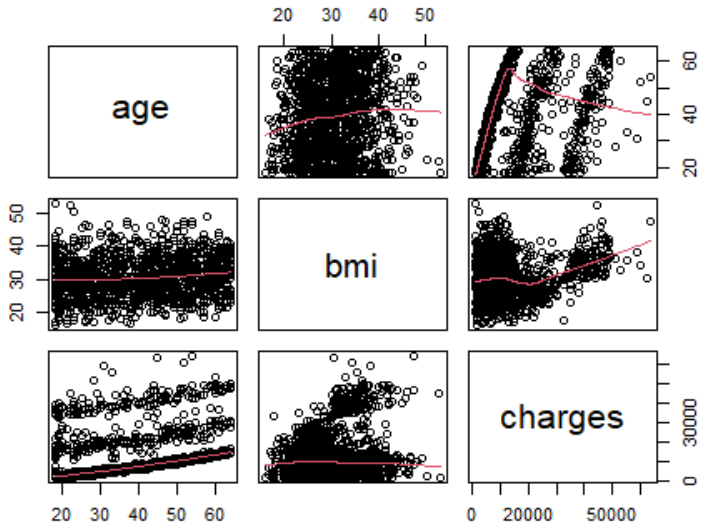
The “smoker” variable seems to have a strong relationship with charges. Looking at the boxplot for “smoker”, patients who smoked pay significantly more in medical costs.

```
region = as.factor(region)
plot(region, charges, main = "Boxplot of Region and Charges", xlab =
"region", ylab = "charges")
```



The “region” variable does not seem very significant. I do not think this variable can help better explain the outcome variable.

```
pairs(insurance[,c(1,3,7)], panel = panel.smooth)
```



To find the correlation between y and each x variable, I used the “pairs” command which gives me their plots along with correlations between each x variable. The “pairs” plot between bmi and charges suggest that higher values of bmi would lead to higher charges, which makes sense because high values of bmi are less healthy. Also the age vs charges graph shows the two variables have a postive association. As Age increases, so does medical charges. We expect this to be the case because people tend to develop more health problems as the grow older, causing their medical costs to increase.

Linear Regressions

There are seven total variables in my dataset with “charges” being my dependent variable. One of the predictive covariates is “region” which is a qualitative predictor and takes values of northwest, northeast, southwest, and southeast. I do not think a person’s region should have an impact on their medical charges so I will not be including this variable in my regressions. This leaves my with five predictors that I will use to explain charges.

```
lm.fit1 = lm(charges~smoker, data = insurance)
summary(lm.fit1)

##
## Call:
## lm(formula = charges ~ smoker, data = insurance)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -19221  -5042   -919   3705  31720
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8434.3      229.0   36.83   <2e-16 ***
## smokeryes    23616.0      506.1   46.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7470 on 1336 degrees of freedom
## Multiple R-squared:  0.6198, Adjusted R-squared:  0.6195
## F-statistic:  2178 on 1 and 1336 DF,  p-value: < 2.2e-16
```

The variable I think will explain charges the most is "smoker". This is a dummy variable that takes a value of yes or no if the person smokes. Smoking can lead to many detrimental health issues which would lead to higher medical costs. I expect people who smoke to pay more in charges than people who do not smoke. The first linear regression I ran shows that the x variable is statistically significant using a t test. However the Adjusted R-squared value is only 0.6195 so I will add additional x variables to the model.

```
lm.fit2 = lm(charges~smoker+bmi, data = insurance)
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = charges ~ smoker + bmi, data = insurance)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15992.7  -4600.2   -802.4   3636.2  30677.8
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3459.10     998.28  -3.465 0.000547 ***
## smokeryes   23593.98     480.18  49.136  < 2e-16 ***
## bmi           388.02      31.79  12.207  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7088 on 1335 degrees of freedom
## Multiple R-squared:  0.6579, Adjusted R-squared:  0.6574
## F-statistic:  1284 on 2 and 1335 DF,  p-value: < 2.2e-16
```

"bmi" is a general measure of a person's height to weight ratio. Adding this variable to the model increased the Adjusted R-squared from 0.6195 to 0.6574, which is a significant increase. The "bmi" variable is also statistically significant using a t-test but its coefficient is not very big. I will continue to include "bmi" in my overall model since it has some impact on "charges" and it increased the Adjusted R-squared.

```
lm.fit3 = lm(charges~smoker+bmi+sex , data = insurance)
summary(lm.fit3)
```

```
##
## Call:
## lm(formula = charges ~ smoker + bmi + sex, data = insurance)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15870.3  -4558.6   -800.6   3671.4  30798.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3353.53    1008.78  -3.324  0.00091 ***
## smokeryes   23620.85     481.66  49.040  < 2e-16 ***
## bmi           389.09      31.83  12.225  < 2e-16 ***
## sexmale      -285.28     389.18  -0.733  0.46366
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7089 on 1334 degrees of freedom
## Multiple R-squared:  0.6581, Adjusted R-squared:  0.6573
## F-statistic: 855.8 on 3 and 1334 DF,  p-value: < 2.2e-16
```

The next variable I included in my model was "sex" which is another dummy variable. The t test statistic for this variable is -0.733 and is not statistically significant even at the 10% level. Including "sex" in my model also lowered the Adjusted R-squared from 0.6574 to 0.6573. I do not think a person's sex should have an effect on how much they pay in medical expenses and the data supports this claim. Moving forward, I will omit the "sex" variable because it does not help explain "charges"

```
lm.fit4 = lm(charges~smoker+bmi+age , data = insurance)
summary(lm.fit4)
```

```
##
## Call:
## lm(formula = charges ~ smoker + bmi + age, data = insurance)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12415.4  -2970.9   -980.5   1480.0  28971.8
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11676.83     937.57  -12.45   <2e-16 ***
## smokeryes    23823.68     412.87   57.70   <2e-16 ***
## bmi            322.62      27.49   11.74   <2e-16 ***
## age            259.55      11.93   21.75   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6092 on 1334 degrees of freedom
```

```
## Multiple R-squared:  0.7475, Adjusted R-squared:  0.7469
## F-statistic:  1316 on 3 and 1334 DF,  p-value: < 2.2e-16
```

I substituted out the "sex" variable for the the "age" variable and this produced much better results. "age" is statistically significant at the 1% level and it also increased the Adjusted R-squared significantly from 0.6574 to 0.7469. It makes sense that "age" affects "charges" because older people generally develop more health problems and need to pay more for medical costs.

```
lm.fit5 = lm(charges~smoker+bmi+age+children , data = insurance)
summary(lm.fit5)

##
## Call:
## lm(formula = charges ~ smoker + bmi + age + children, data = insurance)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11897.9  -2920.8   -986.6   1392.2  29509.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12102.77     941.98 -12.848  < 2e-16 ***
## smokeryes    23811.40     411.22  57.904  < 2e-16 ***
## bmi            321.85      27.38  11.756  < 2e-16 ***
## age            257.85      11.90  21.675  < 2e-16 ***
## children       473.50     137.79   3.436 0.000608 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6068 on 1333 degrees of freedom
## Multiple R-squared:  0.7497, Adjusted R-squared:  0.7489
## F-statistic: 998.1 on 4 and 1333 DF,  p-value: < 2.2e-16
```

The last variable I added to my linear regression is "children" which measures the number of dependents a patient has. This variable is statistically significant and it also slightly increases the Adjusted R-squared value to 0.7489. The coefficient of "children" is postive which means that patients with more children have higher medical costs.

Analyzing the Best Regression Model

The fifth linear regression model I created produces the best results. The model includes four predictors that explain "charges" and below is a summary of the model.

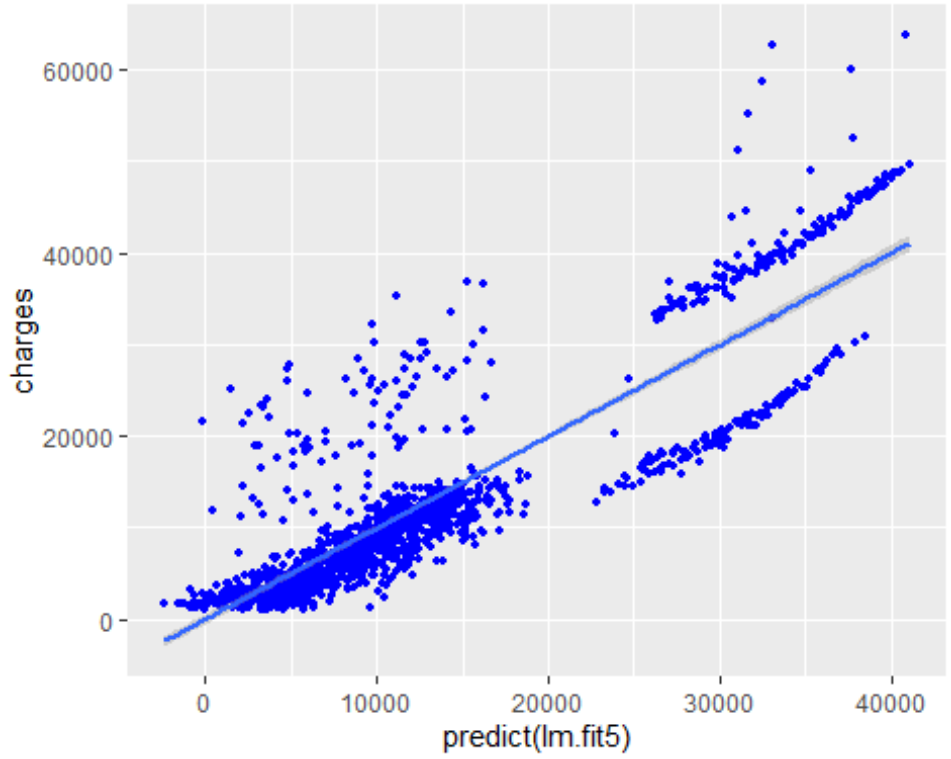
```
summary(lm.fit5)

##
## Call:
## lm(formula = charges ~ smoker + bmi + age + children, data = insurance)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -11897.9  -2920.8   -986.6   1392.2  29509.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12102.77     941.98 -12.848  < 2e-16 ***
## smokeryes    23811.40     411.22  57.904  < 2e-16 ***
## bmi            321.85      27.38  11.756  < 2e-16 ***
## age            257.85      11.90  21.675  < 2e-16 ***
## children       473.50     137.79   3.436 0.000608 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6068 on 1333 degrees of freedom
## Multiple R-squared:  0.7497, Adjusted R-squared:  0.7489
## F-statistic: 998.1 on 4 and 1333 DF,  p-value: < 2.2e-16
```

The four x variables have different levels of effect on medical charges and we can interpret each coefficient. The “smoker” variable has, by far, the largest effect on “charges”. If a person is a smoker, holding everthing else in the equation constant, their medical costs are $23,811.40 higher than a person who is not a smoker. The interpretation of bmi is that for a one unit increase in body mass index, charges go up by $321.85. The interpretation for age is that for a one year increase in age, medical charges increase by $257.85. And if a person has one more child, their charges increase by $473.50.

We can also conclude that all the coefficients are different from zero. The null hypothesis in linear regressions is that the coefficients are zero. In our model, the p-values for each coefficient (using t tests) is virtually zero so we can reject the null hypothesis and conclude that the coefficients have high statistical significance. We can also look at the F test for overall significance. The null hypothesis of the F-test is that all the coefficients equal zero and the alternative is that at least one coefficient is non-zero. The p value of the F test is essentially zero therefore we conclude that at least one of the coefficients is different from zero.

```
ggplot(insurance, aes(x=predict(lm.fit5), y=charges)) +
  geom_point(color='blue', size = 1) +
  geom_smooth(method='lm', formula= y~x)
```



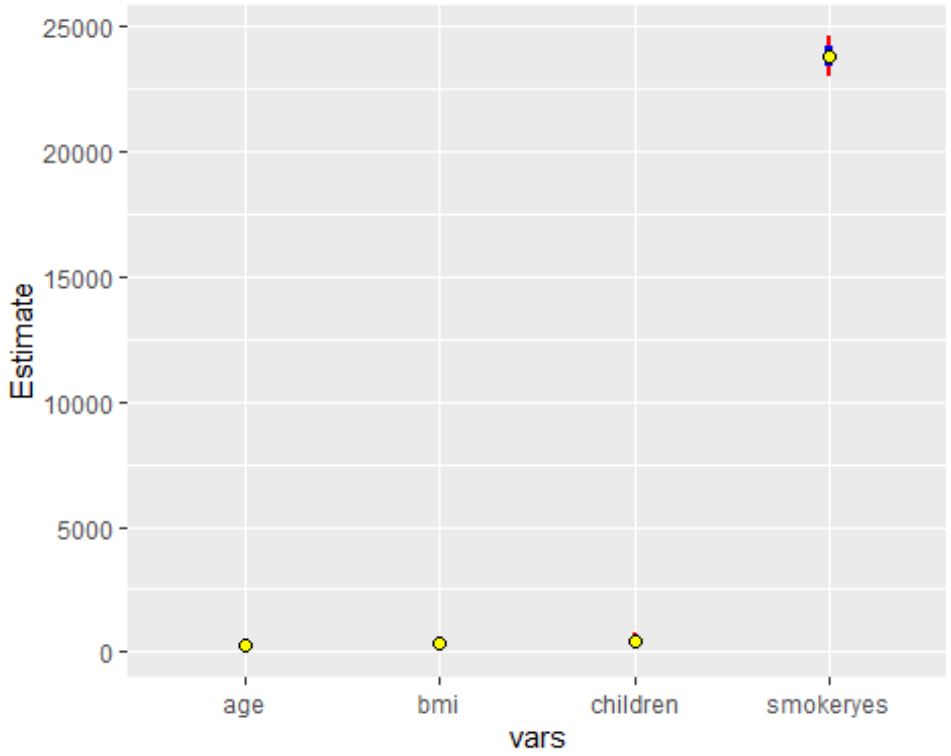

```
  labs(y="actual value", x="fitted value")
```

```
## $y
## [1] "actual value"
##
## $x
## [1] "fitted value"
##
## attr(,"class")
## [1] "labels"
```

Above is the plot that shows the actual value for "charges" vs. their predicted value. The plot seems normal for values below $20,000 but there are some strange patterns in the plot for charges that are greater than $20,000.

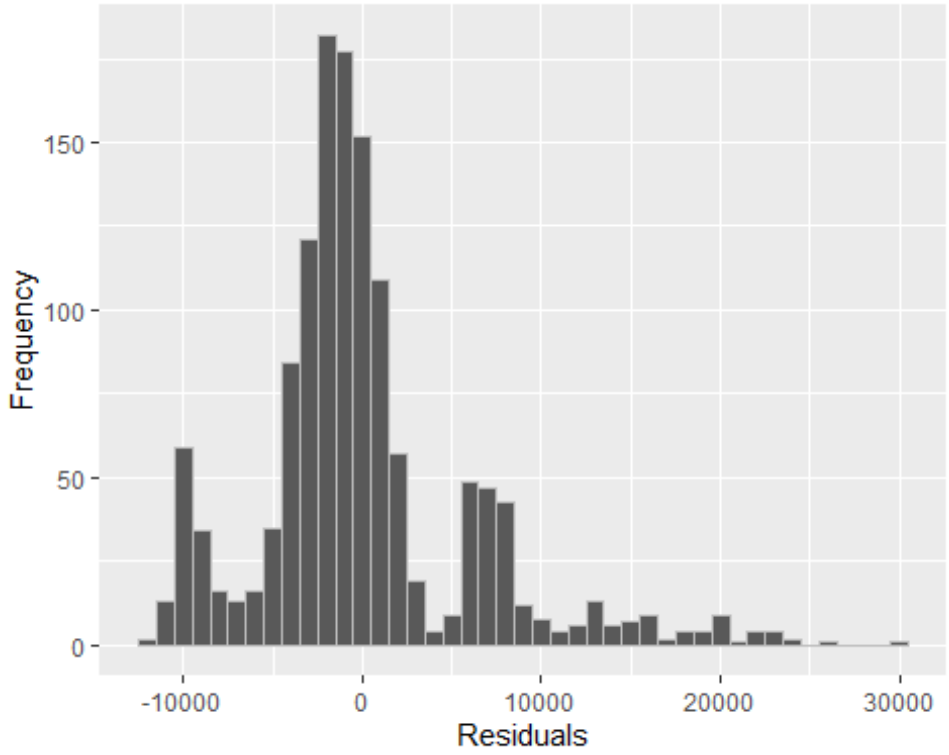
```
lm.summary = summary(lm.fit5)

coefs = as.data.frame(lm.summary$coefficients[-1,1:2]) # -1 is to exclude the
intercept
names(coefs)[2] = "se"
coefs$vars = rownames(coefs)
ggplot(coefs, aes(vars, Estimate)) +
geom_errorbar(aes(ymin=Estimate - 1.96*se, ymax=Estimate + 1.96*se), lwd=1,
```

```
colour="red", width=0) +
geom_errorbar(aes(ymin=Estimate - se, ymax=Estimate + se), lwd=1.5,
colour="blue", width=0) +
geom_point(size=2, pch=21, fill="yellow")
```



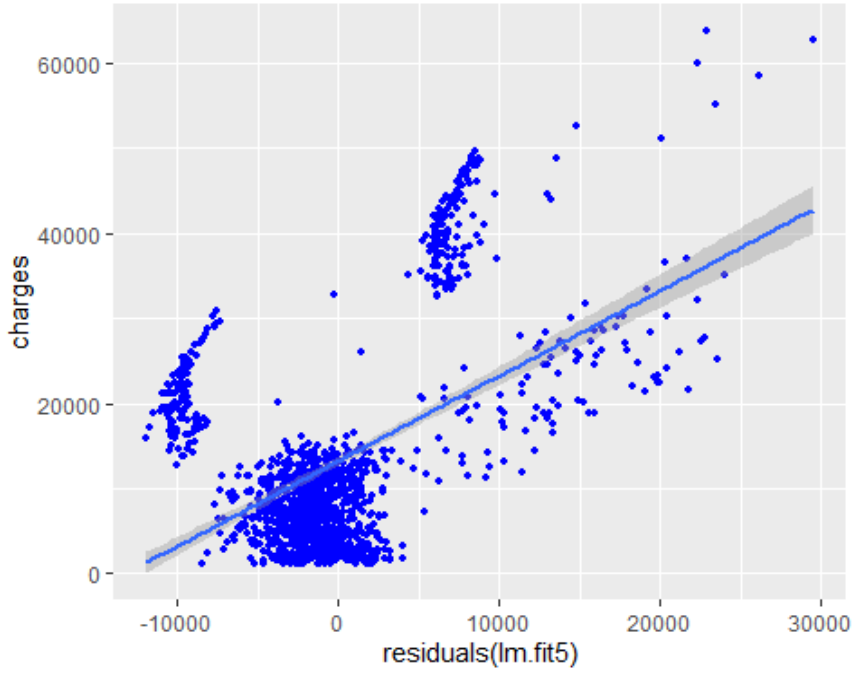
The vars vs. estimate plot shows the four coefficients with their 1.96 standard deviation confidence interval. Since the "smoker" coefficient is so much larger than the other coefficients, the scale on the left side of the graph is very large. As discussed earlier in the report, all of the coefficients are statistically significant.

```
ggplot(insurance) +
  labs(y="Frequency", x="Residuals") +
  geom_histogram(aes(x=residuals(lm.fit5)),binwidth = 1000, colour='grey')
```



The histogram above shows the frequency of the residuals of the linear model. Most of the data is normally distributed around zero.

```
ggplot(insurance, aes(x=residuals(lm.fit5), y=charges)) +
  geom_point(color='blue', size = 1) +
  geom_smooth(method='lm', formula= y~x)
```



```
  labs(y="actual value", x="residuals")
```

```
## $y
## [1] "actual value"
##
## $x
## [1] "residuals"
##
## attr(,"class")
## [1] "labels"
```

This last plot shows the residuals vs the actual value. The positive residuals are larger in magnitude than the negative residuals

Ridge

Splitting I will use these training and testing sets for future models

```
set.seed(45)
train = insurance %>% sample_frac(0.8)
test = insurance %>% setdiff(train)
#both your X_train and X_test should be in matrix format.
x_train = model.matrix(charges~., train)[,-1]
x_test = model.matrix(charges~., test)[,-1]
y_train = train$charges
y_test = test$charges
```

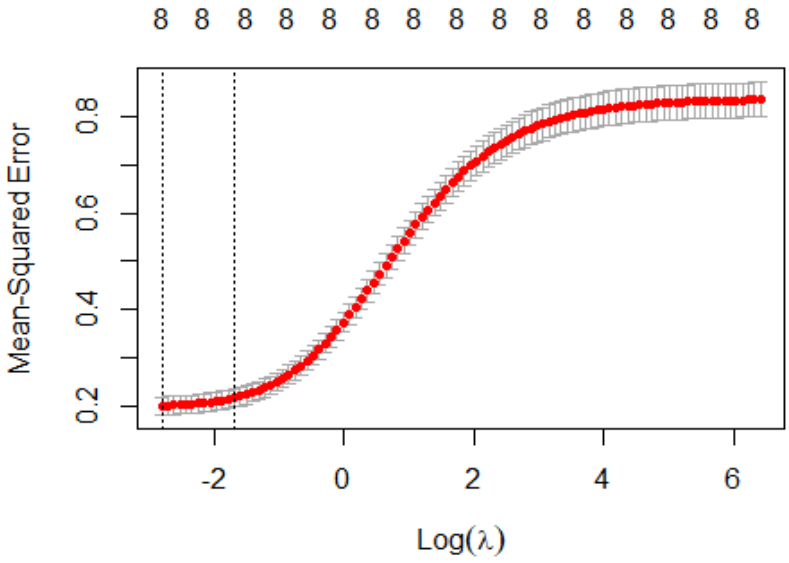
Ridge Regression

```
#Fit ridge regression model on training data, alpha = 0 since ridge
cv.out = cv.glmnet(x_train, log(y_train), alpha = 0)
# Select lamda that is within 1 standard error of the minimum lambda
bestlam = cv.out$lambda.min
bestlam
```

```
## [1] 0.06056632
```

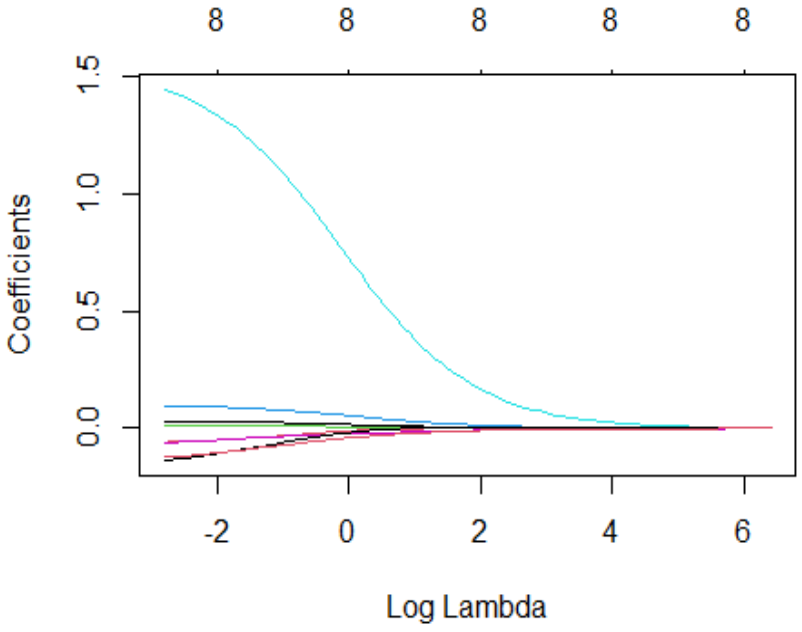
The λ that is within 1 standard error of the minimum λ is 0.0606

```
plot(cv.out)
```



The graph shows the relationship between the cross-validation error and log of λ which is selected, and also shows the minimum λ and λ within 1 standard error of the min.

```
out = glmnet(x_train, log(y_train), alpha = 0)
plot(out, xvar = "lambda")
```



This graph shows the coefficients vary as log of lambda changes. All the coefficients shrink to 0 when log lambda is about 6.

```r
#the coefficients correspondent with the chosen lambda
predict(out, type = "coefficients", s = bestlam)
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                             1
## (Intercept)      7.18837694
## age              0.03246469
## sexmale         -0.05851938
## bmi              0.01127942
## children         0.09725131
## smokeryes        1.44460419
## regionnorthwest -0.06132358
## regionsoutheast -0.13520757
## regionsouthwest -0.12364329
```

None of the predicted coefficients are exactly zero. Note that the coefficient of smokeryes is very large indicating that it is important in predicting charges.

```r
# Using the best lambda to predict test data
ridge_pred = predict(out, s = bestlam, newx = x_test)
ridge_mse = mean((exp(ridge_pred) - y_test)^2)
ridge_mse
```

```
## [1] 71457279
```

```r
# Full OLS model
ols_f = lm(log(charges)~., data = train)
ols_pred = predict(ols_f, test)
ols_mse = mean((exp(ols_pred) - y_test)^2) #MSE
ols_mse
```

```
## [1] 86503360
```

```r
# Best model from report 2
ols_best = lm(log(charges) ~ smoker + bmi + age + children, data = train)
ols_pred = predict(ols_best, test)
ols_mse = mean((exp(ols_pred) - y_test)^2) #MSE
ols_mse
```

```
## [1] 82755401
```

After finding the MSE for the ridge model, full OLS model, and the best model from report 2, I found that the Ridge regression has the best performance as it has the lowest MSE.

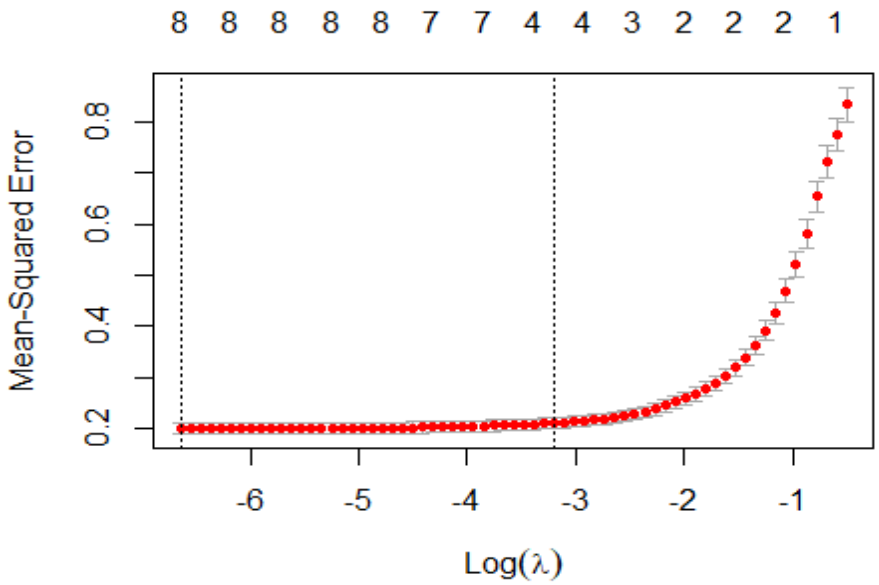
Lasso

```r
#Fit lasso regression model on training data
cv.out_l = cv.glmnet(x_train, log(y_train), alpha = 1)
# Select lamda that is within 1 standard error of the minimum lambda
bestlam_l = cv.out_l$lambda.1se
bestlam_l
```

```
## [1] 0.04078627
```

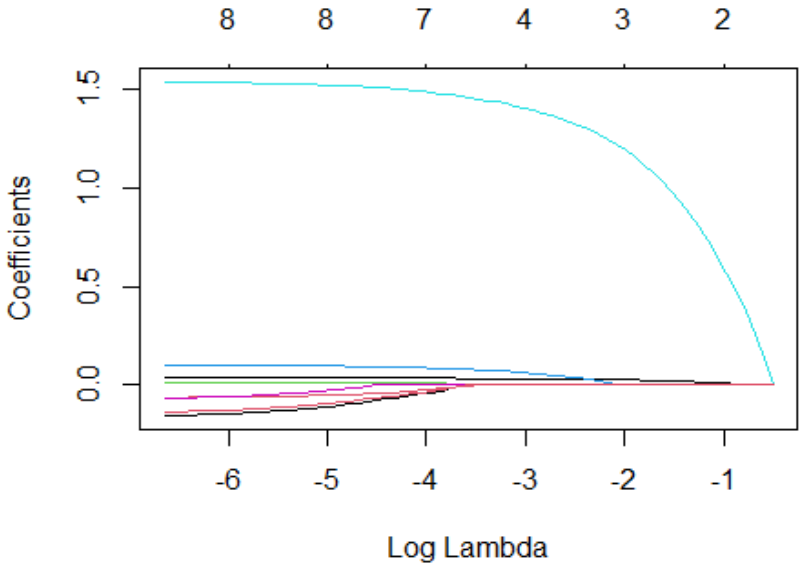
The λ that is within 1 standard error of the minimum λ is 0.0448

```
plot(cv.out_l) #plot of training MSE as a function of lambda
```



The graph shows the relationship between the cross-validation error and log of λ which is selected, and also shows the minimum λ and λ within 1 standard error of the min.

```
out_l = glmnet(x_train, log(y_train), alpha = 1)
plot(out_l, xvar = "lambda") #now coefficients vary with lambda
```



As the log of lambda varies, some of the coefficients are exactly equal to 0, while others are nonzero for large negative values but eventually converge to zero.

```
#the coefficients correspondent with the chosen lambda
predict(out_l, type = "coefficients", s = bestlam_l)
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                              1
## (Intercept)     7.358237033
## age             0.032223908
## sexmale         .
## bmi             0.003498712
## children        0.069267370
## smokeryes       1.429822363
## regionnorthwest .
## regionsoutheast .
## regionsouthwest .
```

Now in the Lasso Regression, 4 of the 9 coefficients are exactly zero. We can omit the zero coefficients from our model.

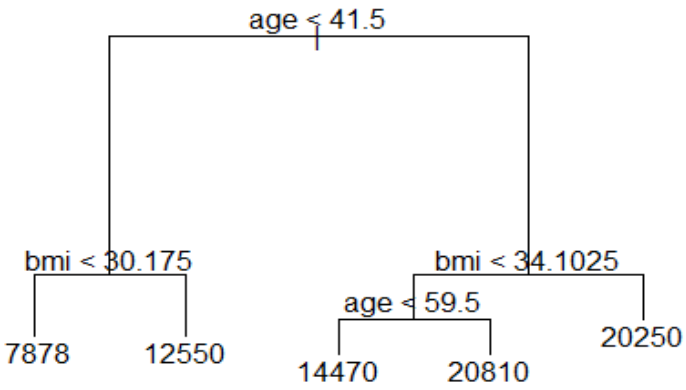
```
lasso_pred = predict(out_l, s = bestlam_l, newx = x_test)
# Calculate the MSE for Lasso
lasso_mse = mean((exp(lasso_pred) - y_test)^2)
lasso_mse
```

```
## [1] 73220186
```

After finding the MSE for Lasso regression, Ridge Regression, and linear regressions, the model that performed the best is the Ridge regression. It has an MSE that is significantly lower than the linear regression and slightly lower than the Lasso regression.

Decision Tree

```
big_tree = tree(charges~., train)
plot(big_tree)
text(big_tree, pretty = 0)
```

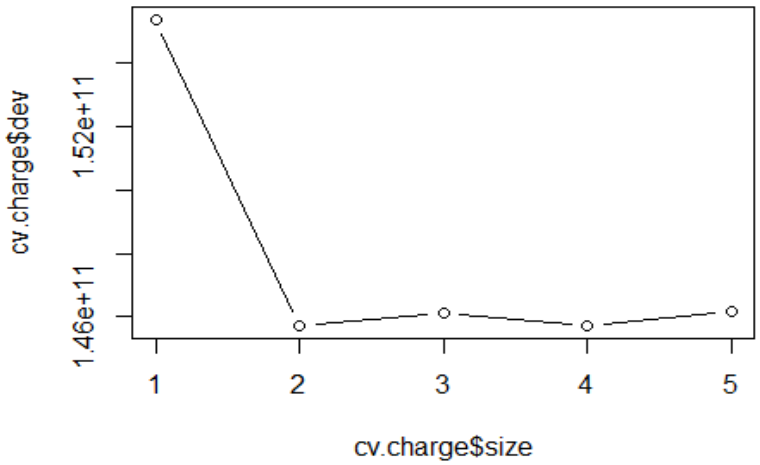


The tree has a depth of 3 and contains 5 leaves.

```
big_tree_pred = predict(big_tree, test)
tree_MSE = mean((big_tree_pred - y_test)^2)
tree_MSE

## [1] 135906413
```

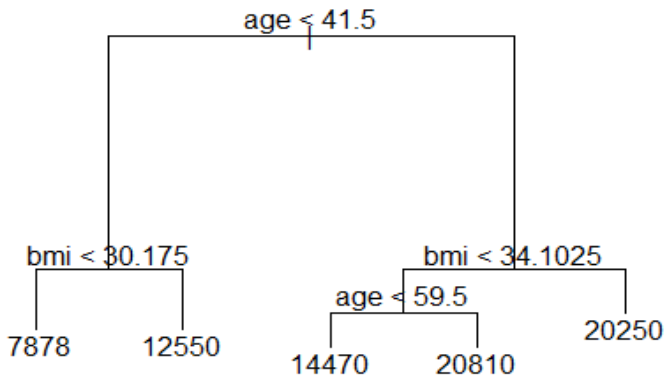
Now to prune the tree

```
cv.charge = cv.tree(big_tree)
plot(cv.charge$size, cv.charge$dev, type = 'b')
```



The graph illustrates a 5 terminal node tree that is selectedby cross-validation

```
prune_charge = prune.tree(big_tree,best = 5)
plot(prune_charge)
text(prune_charge, pretty = 0)
```



The first split uses the variable smoker. For those who did not smoke, it split by the age with a threshold of 42.5 years. For those who did smoke, it split by a BMI threshold of 30.01. Those who had BMI > 30.01 were split again by age at a threshold of 43.5

```
tree_pred = predict(prune_charge,test)
tree_mse = mean((tree_pred-y_test)^2)
tree_mse

## [1] 135906413
```

The tree regression has the best performance out of all the models thus far. It has by far the lowest MSE. The second lowest MSE is for the Ridge regression.

Bagging

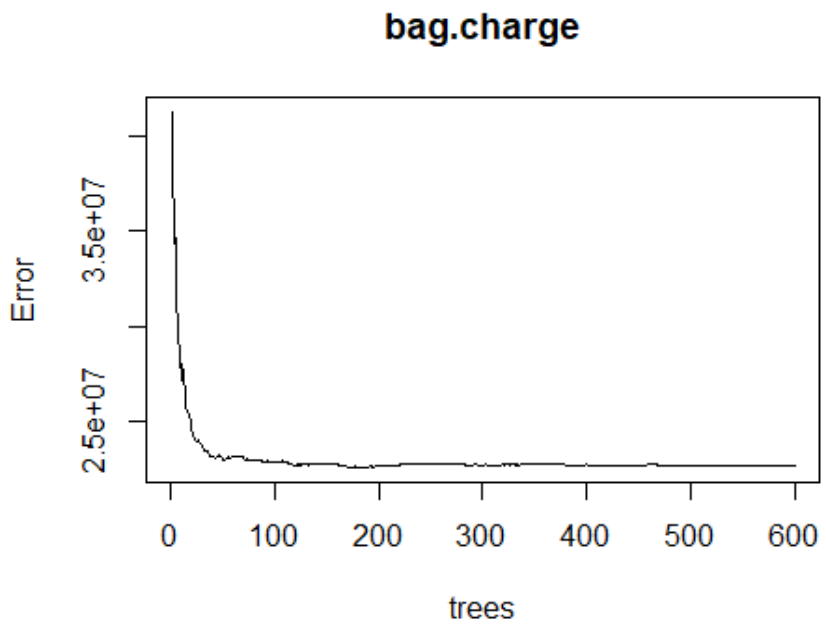
Out-of-Bag Error Rate vs Number of Trees

```
bag.charge = randomForest(charges ~., train, mtry = ncol(train)-1, ntree =
600, importance = TRUE, do.trace = 100)

##      |      Out-of-bag   |
## Tree |      MSE  %Var(y) |
##  100 | 2.286e+07    15.79 |
##  200 | 2.264e+07    15.63 |
##  300 | 2.271e+07    15.68 |
##  400 | 2.271e+07    15.68 |
```

```
##   500 | 2.264e+07    15.63 |
##   600 | 2.266e+07    15.65 |
```

The argument mtry = ncol(train) - 1 indicates that all 5 predictors should be considered for each split of the tree. I set the model to generate 600 trees and trace the out-of-bag error for every 100 trees.

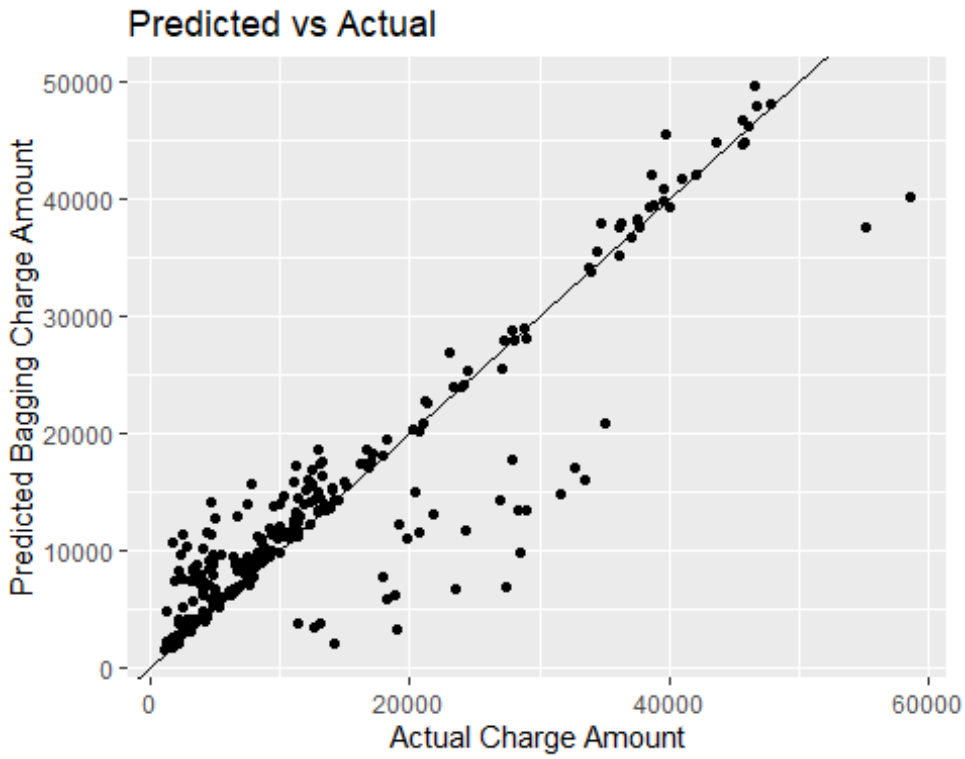
```
plot(bag.charge)
```



The graph above shows the relationship between out-of-bag error with number of trees with up to 600 trees. The error decreases the most in the first 70 trees, then it decreases very gradually.

Predicted Y vs Acutal Y

```
bag_pred = predict(bag.charge, newdata = test)
ggplot() +
  geom_point(aes(x=test$charges,y=bag_pred))+
  labs(title = "Predicted vs Actual")+
  geom_abline() +
  labs(x = "Actual Charge Amount", y = "Predicted Bagging Charge Amount")
```



The Actual vs Predicted model generally follows a linear trend. The Bagging model does underestimate some of the actual charge amounts, as seen by the points that are to the right of abline.

Comparing MSE to Previous Models

```
bag_mse = mean((bag_pred - test$charges)^2)
bag_mse
```

```
## [1] 24968378
```

The bagging model performs significantly better than the Lasso and Ridge regresssions. The MSE for the bagging model is less than half the value of the Ridge model.

Importance Matrix

```
importance(bag.charge)
```

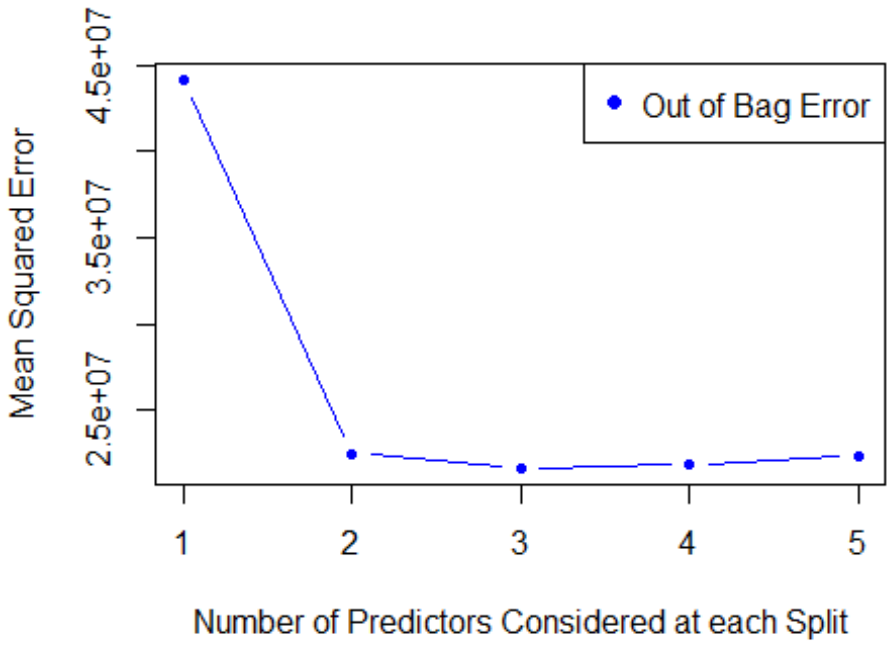
```
##             %IncMSE IncNodePurity
## age      114.952753   18998315447
## sex       -3.441706     571941812
## bmi      126.786539   31681384445
## children  30.293703    2909012206
## smoker   235.774940   95469596651
## region     6.315887    1612941642
```

In the importance matrix, there are three large %IncMSE values compared to the rest. This means that these three variables (age, bmi, and smoker) are important in predicting the charge than the other variables.

Random Forest Regression

Out-Of-Bag Error Rate vs. Number of Predictors

```
# Set mtry using hyperparamter tuning
oob.err = double(5)
test.err = double(5)

#mtry is no of Variables randomly chosen at each split
for(mtry in 1:5)
{
  rf=randomForest(charges ~ . , data = train, mtry=mtry, ntree=400)
  oob.err[mtry] = rf$mse[400] #Error of all Trees fitted on training

  pred<-predict(rf,test) #Predictions on Test Set for each Tree
  test.err[mtry]= mean( (pred - test$charges)^2) # "Test" Mean Squared Error

}

matplot(1:mtry , oob.err, pch=20 , col="blue",type="b",ylab="Mean Squared
Error",xlab="Number of Predictors Considered at each Split")
legend("topright",legend=c("Out of Bag Error"),pch=19, col=c("blue"))
```



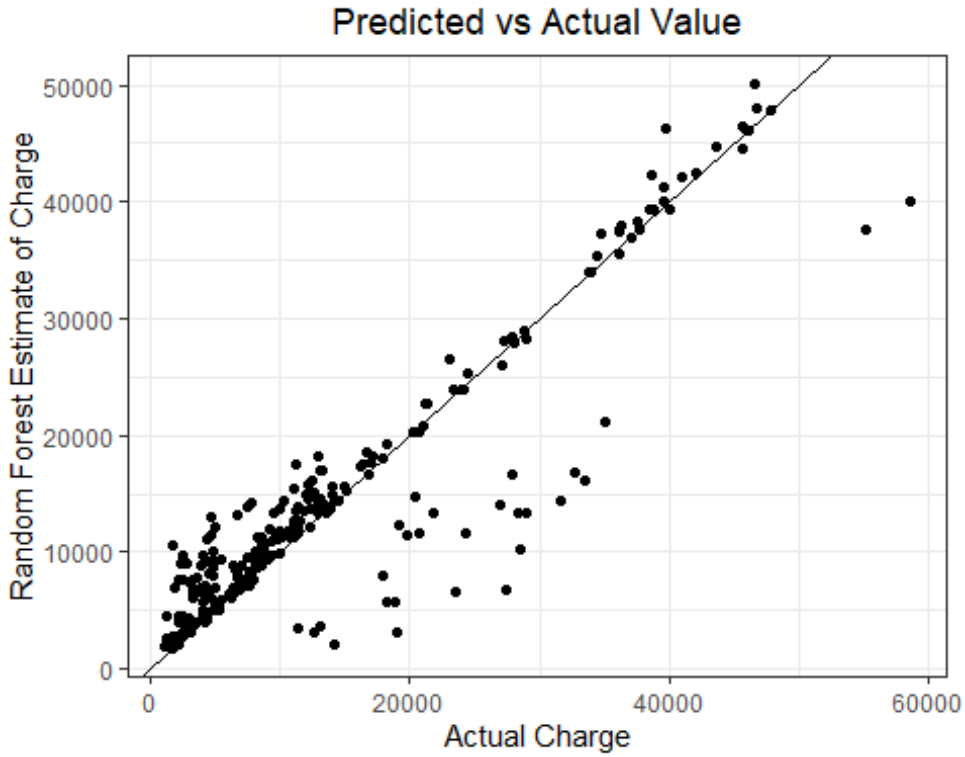
The MSE dramatically drops after 2 predictors. The Minimum MSE is achieved with 3 predictors.

Tuning Predictors at Each Split

```
registerDoMC(4)
rf.charge <- foreach(ntree=rep(40, 10), .combine=randomForest::combine,
.multicombine=TRUE, .packages='randomForest') %dopar% {
randomForest(charges ~., train, mtry = 5,
ntree = ntree, importance = TRUE)}
```

Predicted vs. Actual Charge

```
rf_pred = predict(rf.charge, newdata = test)
ggplot() +
  geom_point(aes(x=test$charges, y = rf_pred))+
  labs(title = "Predicted vs Actual Value")+
  geom_abline() +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(x = "Actual Charge", y = "Random Forest Estimate of Charge")
```



The graph above shows that Random Forest Estimate of Charges does a good job in predicting the actual value of charge for smaller numbers. It is less accurate for charges ranging from $20,000-$30,000.

Comparing RF MSE to Other Model MSEs

```
rf_mse = mean((rf_pred - test$charges)^2)
rf_mse

## [1] 24483881
```

The MSE of random forest of is much lower than the one for bagging. The RF MSE performs significantly better than all the other models when looking at MSE values.

Importance Matrix

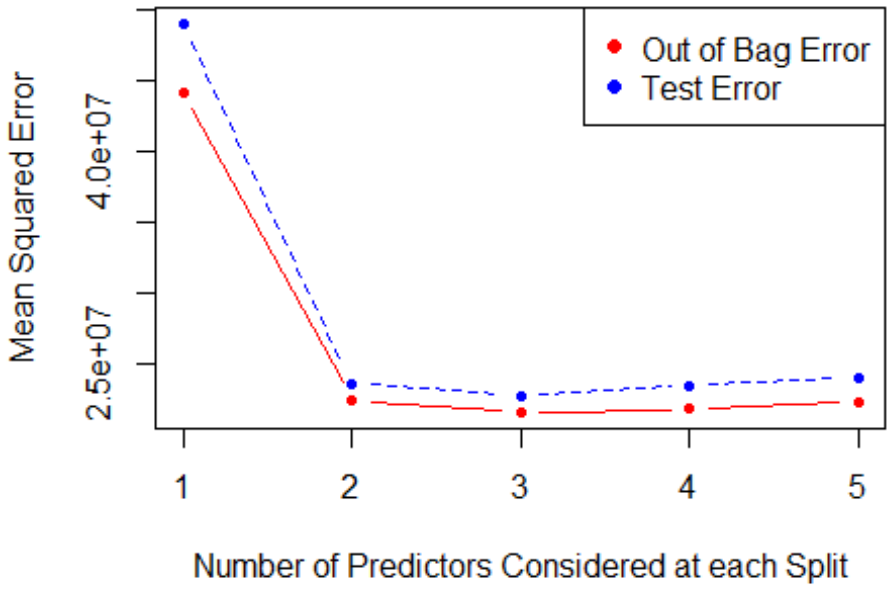
```
importance(rf.charge)
```

```
##             %IncMSE IncNodePurity
## age      31.4331357   19670358895
## sex      -0.7641188     652046742
## bmi      32.9927202   30677913760
## children  7.6980264    2775653289
## smoker   62.9203973   95494087223
## region    2.1390586    1718984409
```

The matrix above shows that the predictors of age, bmi, and smoker are important in predicting the charge of medical costs.

Test and Out-of-Bag Error vs mtry

```
matplot(1:mtry, cbind(oob.err,test.err), pch = 20,
col=c("red","blue"),type="b",ylab="Mean Squared Error",xlab="Number of
Predictors Considered at each Split")
legend("topright",legend=c("Out of Bag Error","Test Error"),pch=19,
col=c("red","blue"))
```

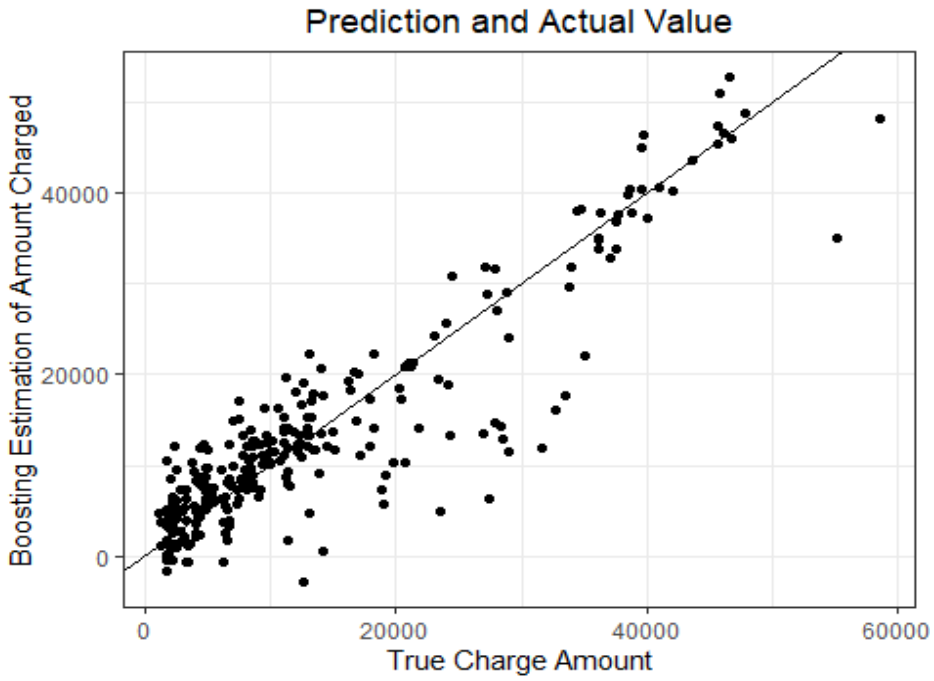


The OOB Eror and Test Error have identical distributions, with the OOB error having slightly higher MSE for each level of predictors

Boosting

Predicted vs Actual Charge

```
# first need to convert columns to from character type to factors in order to
use gbm function
train1 = train
train1$sex = as.factor(train1$sex)
train1$smoker = as.factor(train1$smoker)
train1$region = as.factor(train1$region)

test1 = test
test1$sex = as.factor(test1$sex)
test1$smoker = as.factor(test1$smoker)
test1$region = as.factor(test1$region)

boost.charge = gbm(charges~.,
                   data = train1,
                   distribution = "gaussian",
                   n.trees = 5000, interaction.depth = 4,
                   )
boost_pred = predict(boost.charge, test1, n.trees = 5000)
ggplot() +
geom_point(aes(x = test$charges, y = boost_pred)) +
labs(title = "Prediction and Actual Value") +
theme_bw() +
theme(plot.title = element_text(hjust = 0.5)) +
geom_abline() +
labs(x="True Charge Amount", y="Boosting Estimation of Amount Charged")
```



The graph above shows that the boosting model does a good job in predicting the true charge for charges under $20,000. For charges larger than this vale, there is less accuracy.This model performs better than the Random Forest model since the data points are much closer to the trend line and there is less error.

Comparing MSE of Boost Model to Other Models

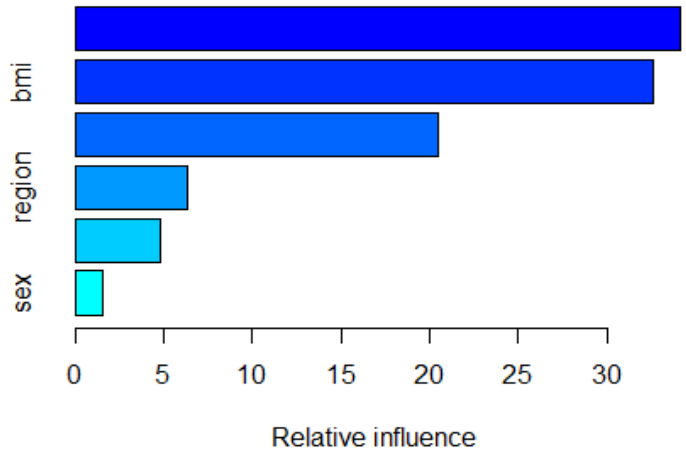
```
boost_mse = mean((boost_pred - test1$charges)^2)
boost_mse
```

```
## [1] 28589133
```

The MSE of the Boost Model performs worse than the Random Forest model, but better than thre tree, lasso, OLS, and ridge models. The Random Forest model stil has the lowest MSE yet.

Importance Matrix

```
# importance(boost.charge)
summary(boost.charge)
```



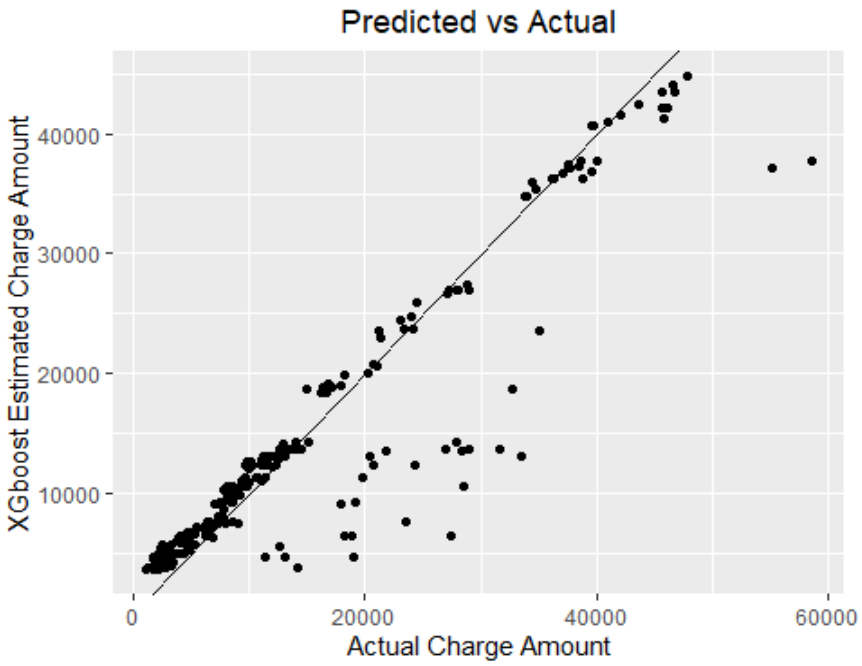

```
##                var   rel.inf
## smoker     smoker 34.174586
## bmi           bmi 32.629402
## age           age 20.500725
## region     region  6.364008
## children children  4.766146
## sex           sex  1.565133
```

Looking at the Relative Influence Graph, the three most important variables in this model are smoker, bmi, and age. In the previous models, the smoker variable was by far the most important variable. It is interesting to note that in thhis model, the variable bmi as nearly the same relative infuence as smoker.

XGboost

Predicted vs Actual Charge

```
# the x_train, y_train, and x_test are already in proper matrix format
dtrain = xgb.DMatrix(data = x_train, label = train$charges)
xgb.charge = xgboost(data=dtrain,
                     max_depth=2,
                     eta = 0.1,
                     nrounds=40, # max number of boosting iterations (trees)
                     lambda=0,
                     print_every_n = 10,
                     objective="reg:linear")

## [1]  train-rmse:16196.955078
## [11] train-rmse:7279.981934
## [21] train-rmse:5051.121094
## [31] train-rmse:4535.374512
## [40] train-rmse:4389.198242

xgb_pred = predict(xgb.charge, x_test)
ggplot() +
  geom_point(aes(x = test$charges, y = xgb_pred)) +
  labs(title = "Predicted vs Actual") +
  theme(plot.title = element_text(hjust = 0.5)) +
  geom_abline() +
  labs(x="Actual Charge Amount", y="XGboost Estimated Charge Amount")
```



The spread of the data points above resembles the graphs for Bagging and Random Forest Models. The predictions above seem less accurate for larger charge amounts and seem more accurate for charge values ranging from $0 to $20,000.

MSE Compairison

```
xgb_mse = mean((xgb_pred - test$charges)^2)
xgb_mse
```

```
## [1] 21192091
```

As I noticed from the Predicted vs Actual graph, the XGboost model does not perform as well as the others. The MSE value of XGboost model is higher than the MSE value for the Boost model which had an MSE.

Importance Matrix

```
xgb.importance(colnames(x_train), model = xgb.charge)
```

```
##      Feature        Gain      Cover Frequency
## 1: smokeryes 0.712478538 0.37032710 0.2916667
## 2:       bmi 0.172640092 0.10445093 0.2083333
## 3:       age 0.107787024 0.43281542 0.3666667
## 4:  children 0.007094346 0.09240654 0.1333333
```

In the matrix above, the important metric to look at is Gain.The smoker variable has the highest Gain value because it is more for generating a prediction than the other variables.

Comparing All Models

Creating a table that has the MSE performance of all the models:

```
mse_table = rbind(ridge_mse, ols_mse, lasso_mse, tree_mse, bag_mse,
rf_mse,xgb_mse, boost_mse)
colnames(mse_table) = "MSE"
mse_table
```

```
##                MSE
## ridge_mse  71457279
## ols_mse    82755401
## lasso_mse  73220186
## tree_mse  135906413
## bag_mse    24968378
## rf_mse     24483881
## xgb_mse    21192091
## boost_mse  28589133
```

After trying many different models, from linear OLS models to Neural Nets, I found the best model to be the XGboost model. The two least accurate models were the Tree model and the Neural Net model. They both had MSE values of greater than 100,000,000. I believe my MSE value for the Neural Net model is not very accurate because although my calculations did produce a MSE value at the end, the graphs did not print out data points. The Ridge, Lasso, and OLS model had similar performances because their MSE values range from 70-80 million range. Finally, the four best models were Bagging, Random Forest, Boost, and XGboost. All of their MSE values were in the 20 million range. The XGboost model had the lowest Mean Squared Error with a value of 21,192,091.

Conclusion

Creating models that can predict a person's medical cost is very important. This information would be particuraly desirable for the health industry and for insurance companies. Insurance companies invest a lot of time, energy, and money in predicting health care costs. Not all of the variables in the Medical Cost data set were important in predicting charges, and the exploratory data analysis at the start of this project gave me some idea of the relationship among the variables. The variables that did not seem significant were sex, children, and region, and this was later confirmed by the various models I created. The variables that repeatedly were significant were age, bmi, and smoker.

I created 5 linear regression models that varied in their number of predictors. To select the best model, I analyzed their R-squared values and choose the highest one. 75% of the variation in charges could be explained by this linear regression model. The other models I later used were not linear regressions so R-squared could not be used to compare models. Instead, Mean Square Error (MSE) was used to evaluate the performance of the models. For the machine learning models, I created training and testing data sets. I used the shrinkage methods of Lasso and Ridge and compared them to the OLS model. The MSE for the shrinkage models was lower than the best linear regression model. A Tree model was also used and subsampled 100 times with boot-straps. The performance of the Tree model was surprisingly poor. Its MSE was even higher the the OLS MSE.

The training sets created earlier were used for all of the following models: Bagging, Random Forest, Boost, and XGboost. These four models all performed significantly better than the first four. As can be seen in the MSE table, ther MSE value was less than half of the first four. In this part of the project, I also created importance matrix which described which independent variables were more valuable in predicting the actual medical charge. The importance matrix for Bagging, Random Forest, and Boosting emphasized that variables age, bmi, and smoker were important in predicting medical charge. Furthermore, the Predicted Charge vs Actual Charge plots had similar pattterns. The models were good at predicting the charge amount was less than $20,0000 and usually underestimated the charge amount. After $20,000, the models were less accurate but did not underestimate the actual charge like before. Of these four models, XGboost performed the best and had the lowest MSE with a value of 21,119,2091.

The data obtained from this project can have powerful real-world implications. This type of information would be highly coveted by billion-dollar industries such as insurance companies, since it would give them a better idea of what to expect from certain customers. They need to be able to predict healthcare costs for forecasting and growing businesses. For example, these models show that whether a person smokes is very important in predicting their medical costs. Insurance companies could use these models to predict the medical costs of their customers, and adjust their insurance rates and premiums accordingly. This project taught me how valuable personal information is and the importance of keeping it private because in the wrong hands, personal information could be sold off and misused. More importantly, I now understand how much we can learn from a data set. Using these machine learning methods, we can gain valuable insights through predictions.