Ensembling Methodologies to Predict Medical Expenses Among Smokers and Non-Smokers

Joel Laskow, Oluwadamilola Owolabi, and Simi Augustine

2024-02-02

# Summary:

The following is a document outlining our analysis of insurance data provided by “Machine Learning in R (2013)”, by Brett Lantz. This document will guide readers through our analysis and the models constructed to predict patient medical expenses per year. Methodlogies included advanced multiple linear regresion, random forest, KNN regression, and ensembling by avaeraging. Final predictions with ensembled models were cross validated by Leave One Out Cross Validation (LOOCV) and K-Fold Cross Validation.

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.2

library(naniar)

## Warning: package 'naniar' was built under R version 4.3.2

library(FNN)

## Warning: package 'FNN' was built under R version 4.3.2

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.3.2

## randomForest 4.7-1.1

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

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(class)

##   
## Attaching package: 'class'

## The following objects are masked from 'package:FNN':  
##   
## knn, knn.cv

library(caret)

## Warning: package 'caret' was built under R version 4.3.2

## Loading required package: lattice

library(tidyverse)

## Warning: package 'dplyr' was built under R version 4.3.2

## Warning: package 'stringr' was built under R version 4.3.2

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.3 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ lubridate 1.9.3 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.0

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::combine() masks randomForest::combine()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ purrr::lift() masks caret::lift()  
## ✖ randomForest::margin() masks ggplot2::margin()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(lattice)  
library(dplyr)  
library(leaps)

## Warning: package 'leaps' was built under R version 4.3.2

library(glmnet)

## Warning: package 'glmnet' was built under R version 4.3.2

## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 4.3.2

##   
## Attaching package: 'Matrix'  
##   
## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack  
##   
## Loaded glmnet 4.1-8

library(caTools)

## Warning: package 'caTools' was built under R version 4.3.2

library(fastDummies)

## Warning: package 'fastDummies' was built under R version 4.3.2

## Thank you for using fastDummies!  
## To acknowledge our work, please cite the package:  
## Kaplan, J. & Schlegel, B. (2023). fastDummies: Fast Creation of Dummy (Binary) Columns and Rows from Categorical Variables. Version 1.7.1. URL: https://github.com/jacobkap/fastDummies, https://jacobkap.github.io/fastDummies/.

library(base)

# Data Pre Processing:

Our dataset conatins 1338 subjects with 7 variables each. There are 4 continuous variables: Age, body mass index (bmi), number of children (children), and yearly medical expenses (charges). The dataset contains 3 categorical variables: gender (sex), smoker status (smoker), and geographical region (region). All patients are located within the United States of America.

insurance <- read.csv("C:/Users/owola/Documents/MY\_COURSES/SPRING\_2024/DS\_6372/Project/Project 1/insurance.csv")  
  
  
summary(insurance)

## age sex bmi children   
## Min. :18.00 Length:1338 Min. :15.96 Min. :0.000   
## 1st Qu.:27.00 Class :character 1st Qu.:26.30 1st Qu.:0.000   
## Median :39.00 Mode :character 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   
## smoker region charges   
## Length:1338 Length:1338 Min. : 1122   
## Class :character Class :character 1st Qu.: 4740   
## Mode :character Mode :character Median : 9382   
## Mean :13270   
## 3rd Qu.:16640   
## Max. :63770

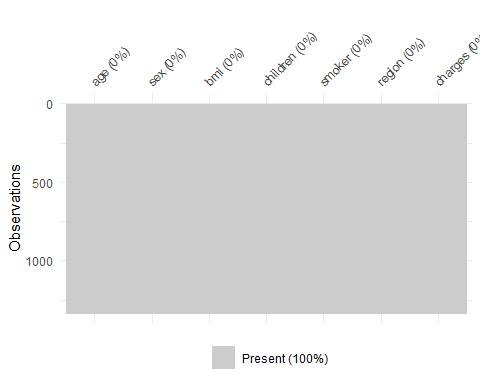
Our dataset can be obtained from the following sources:

* <https://www.packtpub.com/>

or

* <https://www.kaggle.com/datasets/teertha/ushealthinsurancedataset>

vis\_miss(insurance)

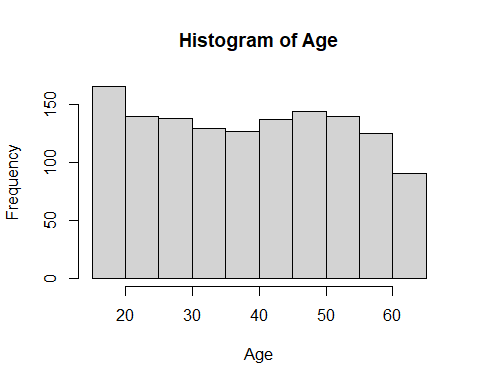


A preliminary assessment of our dataset reveals there are no missing values. This fortunately expedites pre-processing, allowing us to move directly to the data analysis phase.

# Exploratory Data Analysis:

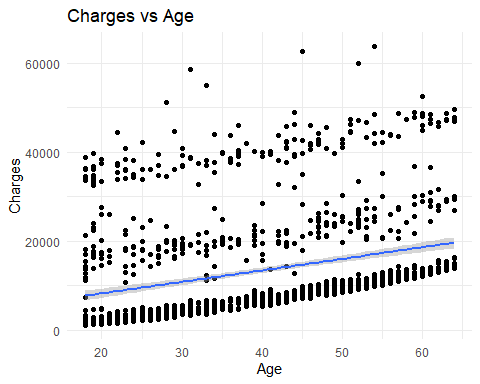
Author: Simi Augustine

hist(insurance$age, main="Histogram of Age", xlab="Age")

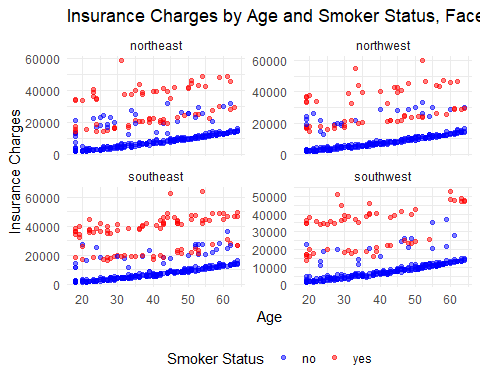


ggplot(insurance, aes(x=age, y=charges)) + geom\_point() +  
 labs(title="Charges vs Age", x="Age", y="Charges") + geom\_smooth(method="lm") + theme\_minimal()

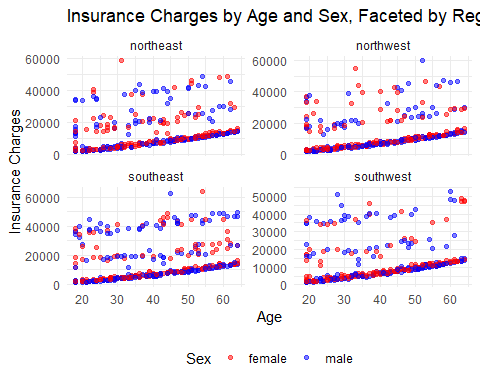
## `geom\_smooth()` using formula = 'y ~ x'



# Create a scatter plot for age vs. insurance charges, color-coded by smoker status  
# and faceted by region  
ggplot(insurance, aes(x = age, y = charges, color = smoker)) +   
 geom\_point(alpha = 0.5) + # Adding transparency to see overlapping points  
 facet\_wrap(~region, scales = "free\_y") + # Creating a separate plot for each region  
 scale\_color\_manual(values = c("yes" = "red", "no" = "blue")) + # Manually assigning colors  
 theme\_minimal() +  
 labs(title = "Insurance Charges by Age and Smoker Status, Faceted by Region",  
 x = "Age",  
 y = "Insurance Charges",  
 color = "Smoker Status") +  
 theme(legend.position = "bottom") # Adjust legend position



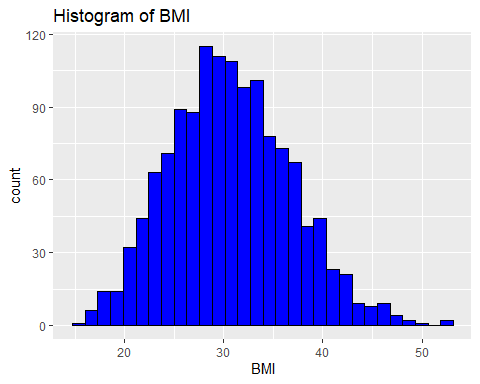
# and faceted by sex  
ggplot(insurance, aes(x = age, y = charges, color = sex)) +   
 geom\_point(alpha = 0.5) + # Adding transparency to see overlapping points  
 facet\_wrap(~region, scales = "free\_y") + # Creating a separate plot for each region  
 scale\_color\_manual(values = c("female" = "red", "male" = "blue")) + # Manually assigning colors  
 theme\_minimal() +  
 labs(title = "Insurance Charges by Age and Sex, Faceted by Region",  
 x = "Age",  
 y = "Insurance Charges",  
 color = "Sex") +  
 theme(legend.position = "bottom") # Adjust legend position



# Histogram for BMI

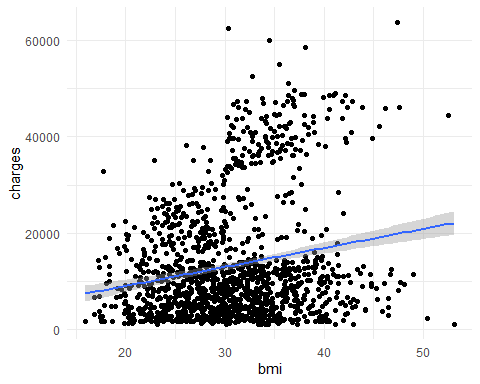
Plot the relationship between Charges and BMI

ggplot(insurance, aes(x = bmi)) +   
 geom\_histogram(bins = 30, fill = "blue", color = "black") +   
 labs(title = "Histogram of BMI", x = "BMI")

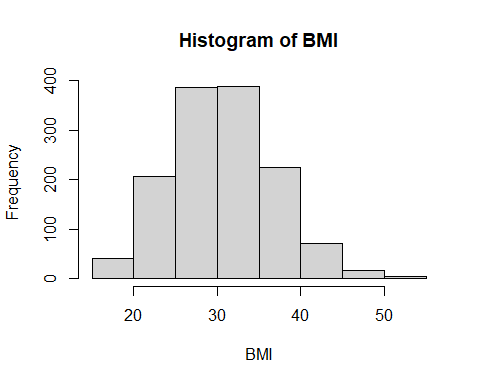


ggplot(insurance, aes(x=bmi, y=charges)) + geom\_point() + geom\_smooth(method="lm") + theme\_minimal()

## `geom\_smooth()` using formula = 'y ~ x'



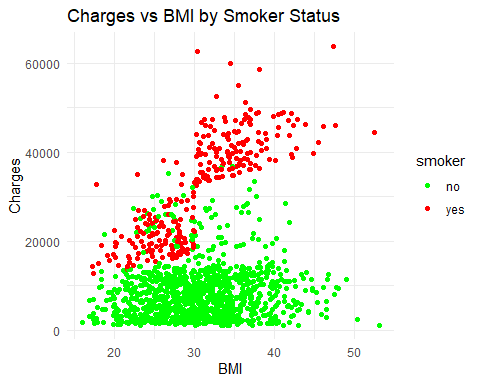
hist(insurance$bmi, main="Histogram of BMI", xlab="BMI")



# BMI, Charges, Smoker

Plot the relationship between Charges and BMI by Smoker status

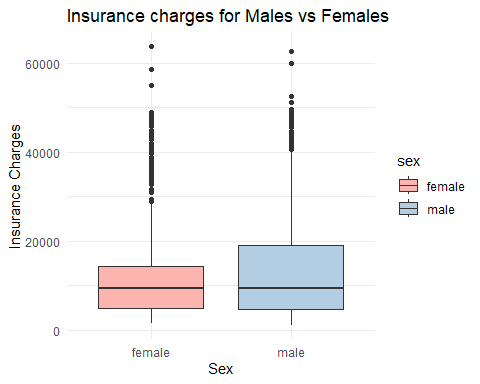
ggplot(insurance, aes(x=bmi, y=charges, color=smoker)) +   
 geom\_point() +   
 labs(title="Charges vs BMI by Smoker Status", x="BMI", y="Charges") +   
 scale\_color\_manual(values=c("yes"="red", "no"="green")) +   
 theme\_minimal()



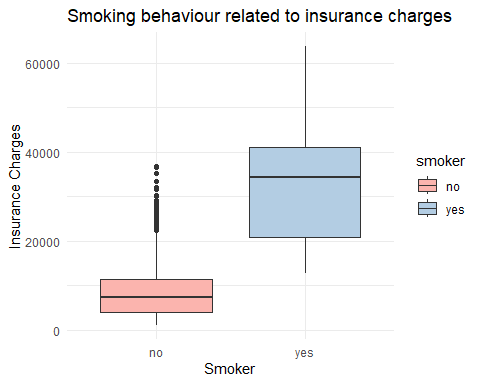
# For categorical variables, use boxplots to visualize the relationship

Smoker, Region, Children

# insurance charges - sex  
ggplot(insurance, aes(x = sex, y = charges, fill = sex)) +   
 geom\_boxplot() +  
 theme\_minimal() +   
 labs(title = "Insurance charges for Males vs Females", x = "Sex", y = "Insurance Charges") +  
 scale\_fill\_brewer(palette = "Pastel1") # Optional: Use a color palette

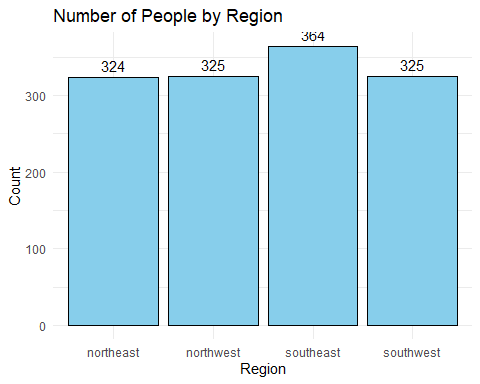


# Smoking behaviour related to insurance charges  
ggplot(insurance, aes(x = smoker, y = charges, fill = smoker)) +   
 geom\_boxplot() +  
 theme\_minimal() +   
 labs(title = "Smoking behaviour related to insurance charges", x = "Smoker", y = "Insurance Charges") +  
 scale\_fill\_brewer(palette = "Pastel1") # Optional: Use a color palette

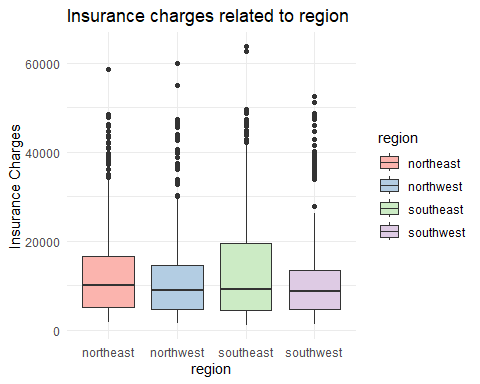


# People distributed by region  
ggplot(insurance, aes(x = region)) +   
 geom\_bar(fill = "skyblue", color = "black") +  
 geom\_text(stat = 'count', aes(label = ..count..), vjust = -0.5) +  
 theme\_minimal() +  
 labs(title = "Number of People by Region", x = "Region", y = "Count") +  
 scale\_x\_discrete(limits = c("northeast", "northwest", "southeast", "southwest"))

## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.  
## ℹ Please use `after\_stat(count)` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

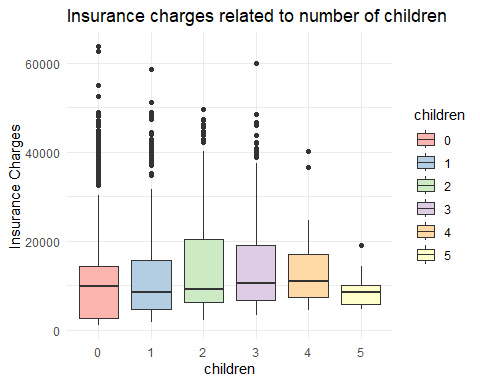


#Insurance charges related to region  
ggplot(insurance, aes(x = region, y = charges, fill = region)) +   
 geom\_boxplot() +  
 theme\_minimal() +   
 labs(title = "Insurance charges related to region", x = "region", y = "Insurance Charges") +  
 scale\_fill\_brewer(palette = "Pastel1") # Optional: Use a color palette

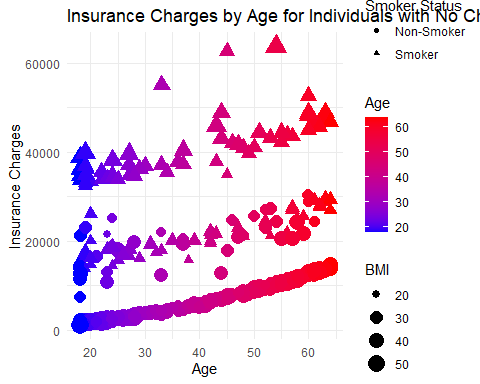


# Children

insurance$children <- as.factor(insurance$children)  
#Insurance charges related to number of children  
ggplot(insurance, aes(x = children, y = charges, fill = children)) +   
 geom\_boxplot() +  
 theme\_minimal() +   
 labs(title = "Insurance charges related to number of children", x = "children", y = "Insurance Charges") +  
 scale\_fill\_brewer(palette = "Pastel1") # Optional: Use a color palette



# Filter for individuals with no children  
insurance\_no\_children <- insurance[insurance$children == 0, ]  
  
# Create a scatter plot, color-coded by age and sized by BMI  
ggplot(insurance\_no\_children, aes(x = age, y = charges, color = age, size = bmi, shape = smoker)) +   
 geom\_point() +  
 scale\_color\_gradient(low = "blue", high = "red") + # Color gradient from blue (younger) to red (older)  
 scale\_size(range = c(1, 6), name = "BMI") + # Adjust the size scale for BMI  
 scale\_shape\_manual(values = c(16, 17), name = "Smoker Status", labels = c("Non-Smoker", "Smoker")) + # Define shapes for smoker status  
 labs(title = "Insurance Charges by Age for Individuals with No Children",  
 x = "Age",  
 y = "Insurance Charges",  
 color = "Age",  
 size = "BMI",  
 shape = "Smoker Status") +  
 theme\_minimal() +  
 theme(legend.position = "right")



# Count the number of children per family  
table(insurance$children)

##   
## 0 1 2 3 4 5   
## 574 324 240 157 25 18

# Objective 1:

Author: Oluwadamilola Owolabi

## Model 1:

## Basic linear regression model

# Getting the csv file  
  
insuranceData <-read.csv("C:/Users/owola/Documents/MY\_COURSES/SPRING\_2024/DS\_6372/Project/Project 1/insurance.csv")  
head(insuranceData)

## 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

insurance <- insuranceData

Data Sorting

#Creating a function for adding new columns  
create\_variable <- function(dataset, Search\_variable, new\_column\_name) {  
 # Create a new column with default value 0  
 dataset[[new\_column\_name]] <- 0  
   
 # Iterate through each row  
 for (i in 1:nrow(dataset)) {  
 # Check if the variable is present in the row  
 if (Search\_variable %in% dataset[i, ]) {  
 dataset[i, new\_column\_name] <- 1  
 }  
 }  
   
 return(dataset)  
}  
  
#adding more variables to the dataset #i forgot to account for sex  
insurance2 <- insurance  
insurance2 <- create\_variable(insurance2, "yes", "is\_smoker") #Adding column to account for presence of smokers  
insurance2 <- create\_variable(insurance2, "no", "not\_smoker") #Adding column to account for absence of smokers  
insurance2 <- create\_variable(insurance2, "southwest", "southwest") #Adding column to account for presence of the southwest region  
insurance2 <- create\_variable(insurance2, "southeast", "southeast") #Adding column to account for presence of the southeast region  
insurance2 <- create\_variable(insurance2, "northwest", "northwest") #Adding column to account for presence of the northwest region  
insurance2 <- create\_variable(insurance2, "northeast", "northeast") #Adding column to account for presence of the northeast region  
insurance2 <- create\_variable(insurance2, "male", "male") #Adding column to account for presence of the male region  
insurance2 <- create\_variable(insurance2, "female", "female") #Adding column to account for presence of the female region  
head(insurance2)

## age sex bmi children smoker region charges is\_smoker not\_smoker  
## 1 19 female 27.900 0 yes southwest 16884.924 1 0  
## 2 18 male 33.770 1 no southeast 1725.552 0 1  
## 3 28 male 33.000 3 no southeast 4449.462 0 1  
## 4 33 male 22.705 0 no northwest 21984.471 0 1  
## 5 32 male 28.880 0 no northwest 3866.855 0 1  
## 6 31 female 25.740 0 no southeast 3756.622 0 1  
## southwest southeast northwest northeast male female  
## 1 1 0 0 0 0 1  
## 2 0 1 0 0 1 0  
## 3 0 1 0 0 1 0  
## 4 0 0 1 0 1 0  
## 5 0 0 1 0 1 0  
## 6 0 1 0 0 0 1

#insurance2 to be used later during MLR an CV ti increase model complexity  
  
#splitting the train and the test dataset  
set.seed(1234)  
index <- createDataPartition(insurance2$charges, p = 0.8, list = FALSE)  
insurance2 <- insurance2[index, ]  
test\_insurance <- insurance2[-index, ]  
  
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

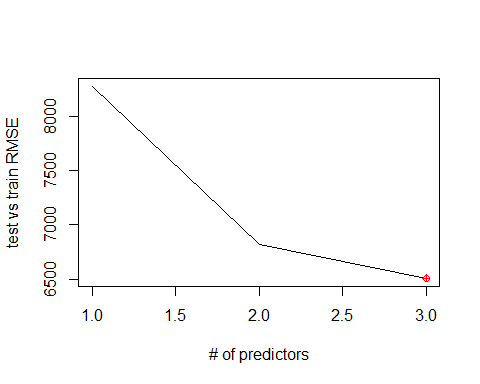
fwd.train=regsubsets(charges ~ .,data=insurance2,method="forward",nvmax=20)

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =  
## force.in, : 8 linear dependencies found

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =  
## force.in, : nvmax reduced to 8

## Warning in rval$lopt[] <- rval$vorder[rval$lopt]: number of items to replace is  
## not a multiple of replacement length

model\_basic = lm(charges ~ .,data=insurance2)  
  
#Creating a prediction function   
predict.regsubsets =function (object , newdata ,id ,...){  
 form=as.formula (object$call [[2]])  
 mat=model.matrix(form ,newdata )  
 coefi=coef(object ,id=id)  
 xvars=names(coefi)  
 mat[,xvars]%\*%coefi  
}  
  
valMSE<-c()  
#note my index, i, is to 20 since that is how many predictors I went up to during fwd selection  
for (i in 1:3){  
 predictions<-predict.regsubsets(object=fwd.train,newdata=test\_insurance,id=i)   
 valMSE[i]<-mean((test\_insurance$charges-predictions)^2)  
}  
  
par(mfrow=c(1, 1))  
plot(1:3,sqrt(valMSE),type="l",xlab="# of predictors",ylab="test vs train RMSE") #ylim=c(11400,11900))  
index<-which(valMSE==min(valMSE))  
points(index,sqrt(valMSE[index]),col="red",pch=10)  
  
trainMSE<-summary(fwd.train)$rss/nrow(test\_insurance)  
lines(1:8,sqrt(trainMSE),lty=3,col="blue")



coef(fwd.train, 3) #Formulae to calculate the validation MSE

## (Intercept) age bmi smokeryes   
## -12044.4689 264.2511 327.1405 23864.7506

summary(fwd.train)

## Subset selection object  
## Call: regsubsets.formula(charges ~ ., data = insurance2, method = "forward",   
## nvmax = 20)  
## 16 Variables (and intercept)  
## Forced in Forced out  
## age FALSE FALSE  
## sexmale FALSE FALSE  
## bmi FALSE FALSE  
## children FALSE FALSE  
## smokeryes FALSE FALSE  
## regionnorthwest FALSE FALSE  
## regionsoutheast FALSE FALSE  
## regionsouthwest FALSE FALSE  
## is\_smoker FALSE FALSE  
## not\_smoker FALSE FALSE  
## southwest FALSE FALSE  
## southeast FALSE FALSE  
## northwest FALSE FALSE  
## northeast FALSE FALSE  
## male FALSE FALSE  
## female FALSE FALSE  
## 1 subsets of each size up to 8  
## Selection Algorithm: forward  
## age sexmale bmi children smokeryes regionnorthwest regionsoutheast  
## 1 ( 1 ) " " " " " " " " "\*" " " " "   
## 2 ( 1 ) "\*" " " " " " " "\*" " " " "   
## 3 ( 1 ) "\*" " " "\*" " " "\*" " " " "   
## 4 ( 1 ) "\*" " " "\*" "\*" "\*" " " " "   
## 5 ( 1 ) "\*" " " "\*" "\*" "\*" " " " "   
## 6 ( 1 ) "\*" " " "\*" "\*" "\*" " " " "   
## 7 ( 1 ) "\*" "\*" "\*" "\*" "\*" " " " "   
## 8 ( 1 ) "\*" "\*" "\*" "\*" "\*" " " " "   
## regionsouthwest is\_smoker not\_smoker southwest southeast northwest  
## 1 ( 1 ) " " " " " " " " " " " "   
## 2 ( 1 ) " " " " " " " " " " " "   
## 3 ( 1 ) " " " " " " " " " " " "   
## 4 ( 1 ) " " " " " " " " " " " "   
## 5 ( 1 ) " " " " " " " " " " " "   
## 6 ( 1 ) " " " " " " " " " " "\*"   
## 7 ( 1 ) " " " " " " " " " " "\*"   
## 8 ( 1 ) "\*" " " " " " " " " "\*"   
## northeast male female  
## 1 ( 1 ) " " " " " "   
## 2 ( 1 ) " " " " " "   
## 3 ( 1 ) " " " " " "   
## 4 ( 1 ) " " " " " "   
## 5 ( 1 ) "\*" " " " "   
## 6 ( 1 ) "\*" " " " "   
## 7 ( 1 ) "\*" " " " "   
## 8 ( 1 ) "\*" " " " "

summary(model\_basic)

##   
## Call:  
## lm(formula = charges ~ ., data = insurance2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11406.0 -2889.4 -950.2 1511.5 24605.7   
##   
## Coefficients: (8 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -12249.92 1112.71 -11.009 < 2e-16 \*\*\*  
## age 260.00 13.24 19.644 < 2e-16 \*\*\*  
## sexmale -447.80 372.84 -1.201 0.230005   
## bmi 359.81 31.97 11.254 < 2e-16 \*\*\*  
## children 533.34 154.47 3.453 0.000577 \*\*\*  
## smokeryes 23914.87 457.46 52.278 < 2e-16 \*\*\*  
## regionnorthwest -542.63 527.39 -1.029 0.303760   
## regionsoutheast -1821.72 536.00 -3.399 0.000702 \*\*\*  
## regionsouthwest -1595.38 535.98 -2.977 0.002981 \*\*   
## is\_smoker NA NA NA NA   
## not\_smoker NA NA NA NA   
## southwest NA NA NA NA   
## southeast NA NA NA NA   
## northwest NA NA NA NA   
## northeast NA NA NA NA   
## male NA NA NA NA   
## female NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6067 on 1063 degrees of freedom  
## Multiple R-squared: 0.755, Adjusted R-squared: 0.7532   
## F-statistic: 409.5 on 8 and 1063 DF, p-value: < 2.2e-16

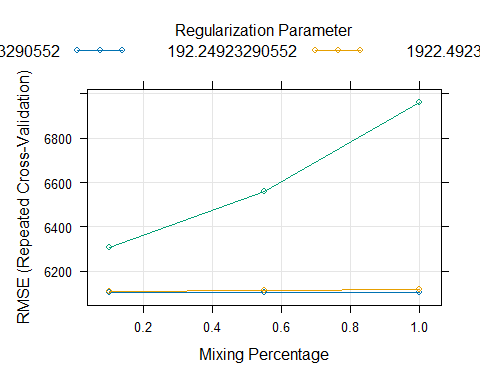
confint(model\_basic)

## 2.5 % 97.5 %  
## (Intercept) -14433.2820 -10066.5636  
## age 234.0299 285.9726  
## sexmale -1179.3881 283.7933  
## bmi 297.0758 422.5491  
## children 230.2316 836.4483  
## smokeryes 23017.2476 24812.4870  
## regionnorthwest -1577.4772 492.2118  
## regionsoutheast -2873.4650 -769.9828  
## regionsouthwest -2647.0813 -543.6833  
## is\_smoker NA NA  
## not\_smoker NA NA  
## southwest NA NA  
## southeast NA NA  
## northwest NA NA  
## northeast NA NA  
## male NA NA  
## female NA NA

#10-fold CV  
fitControl<-trainControl(method="repeatedcv",number=10,repeats=10)   
glmnet.fit<-train(charges ~ (age)^2 + (bmi)^2 + (children)^2 + (is\_smoker)^2,  
 data=insurance2,  
 method="glmnet",  
 trControl=fitControl  
 )  
glmnet.fit

## glmnet   
##   
## 1072 samples  
## 4 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 10 times)   
## Summary of sample sizes: 965, 964, 965, 964, 964, 964, ...   
## Resampling results across tuning parameters:  
##   
## alpha lambda RMSE Rsquared MAE   
## 0.10 19.22492 6107.305 0.7503231 4227.382  
## 0.10 192.24923 6109.484 0.7503190 4237.336  
## 0.10 1922.49233 6306.953 0.7500381 4496.556  
## 0.55 19.22492 6107.435 0.7503129 4221.432  
## 0.55 192.24923 6113.011 0.7502010 4224.729  
## 0.55 1922.49233 6557.642 0.7404533 4691.070  
## 1.00 19.22492 6107.518 0.7503055 4218.552  
## 1.00 192.24923 6119.060 0.7499134 4214.172  
## 1.00 1922.49233 6958.542 0.7149662 5032.536  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final values used for the model were alpha = 0.1 and lambda = 19.22492.

plot(glmnet.fit)



## Complex linear regression model

model.fwd=regsubsets(charges ~ (age)^2 + (bmi)^2 + (children)^2 + (is\_smoker)^2+ (southwest)^2 + (southeast)^2 + (northwest^2) + (northeast)^2 + (male)^2 + (female)^2,data=insurance2,method="forward",nvmax=20)

## Reordering variables and trying again:

summary(model.fwd)$adjr2

## [1] 0.6199283 0.7214066 0.7476771 0.7502115 0.7520078 0.7532596 0.7533590  
## [8] 0.7531690

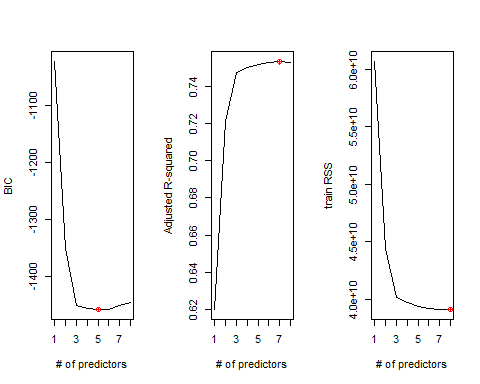
summary(model.fwd)$rss

## [1] 60636467408 44405129218 40180230224 39739417752 39416658972 39180898868  
## [7] 39128340705 39121679927

summary(model.fwd)$bic

## [1] -1024.095 -1351.090 -1451.291 -1456.139 -1457.904 -1457.358 -1451.820  
## [8] -1445.025

par(mfrow=c(1,3))  
bics<-summary(model.fwd)$bic  
plot(1:8,bics,type="l",ylab="BIC",xlab="# of predictors") #got 7 from the forward selection prediction  
index<-which(bics==min(bics))  
points(index,bics[index],col="red",pch=10)  
  
adjr2<-summary(model.fwd)$adjr2  
plot(1:8,adjr2,type="l",ylab="Adjusted R-squared",xlab="# of predictors")  
index<-which(adjr2==max(adjr2))  
points(index,adjr2[index],col="red",pch=10)  
  
rss<-summary(model.fwd)$rss  
plot(1:8,rss,type="l",ylab="train RSS",xlab="# of predictors")  
index<-which(rss==min(rss))  
points(index,rss[index],col="red",pch=10)



coef(model.fwd,8) #looking at coefficints

## (Intercept) age bmi children is\_smoker southeast   
## -13845.3051 260.0013 359.8125 533.3399 23914.8673 -226.3416   
## northwest male northeast   
## 1052.7496 -447.7974 1595.3823

#analysis: out of the 6 coefficients gotten by the forward selction, the BIC chose 4

# Validation testing for complex model

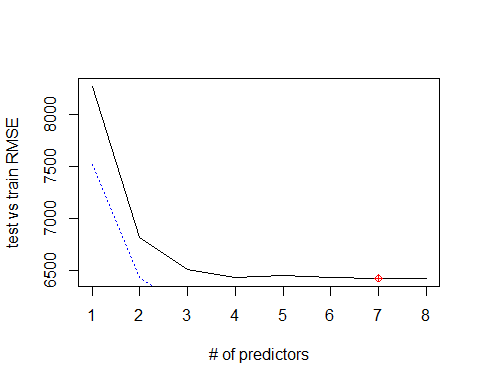
set.seed(1234)  
trainIndex<-createDataPartition(insurance2$charges,p=.8,list=FALSE) #p: proportion of data in train  
  
training<-insurance2[trainIndex,]  
validate<-insurance2[-trainIndex,]  
  
training <- insurance2  
validate <- test\_insurance  
  
fwd.train=regsubsets(charges ~ (age)^2 + (bmi)^2 + (children)^2 + (is\_smoker)^2 + (southwest)^2 + (southeast)^2 + (northwest^2) + (northeast)^2 + (male)^2 + (female)^2,data=training,method="forward",nvmax=20)

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =  
## force.in, : 2 linear dependencies found

## Reordering variables and trying again:

## Warning in rval$lopt[] <- rval$vorder[rval$lopt]: number of items to replace is  
## not a multiple of replacement length

model = lm(charges ~ (age)^2 + (bmi)^2 + (children)^2 + (is\_smoker)^2 ,data=training)  
  
#Creating a prediction function  
predict.regsubsets =function (object , newdata ,id ,...){  
 form=as.formula (object$call [[2]])  
 mat=model.matrix(form ,newdata )  
 coefi=coef(object ,id=id)  
 xvars=names(coefi)  
 mat[,xvars]%\*%coefi  
}  
  
valMSE<-c()  
#note my index, i, is to 20 since that is how many predictors I went up to during fwd selection  
for (i in 1:8){  
 predictions<-predict.regsubsets(object=fwd.train,newdata=validate,id=i)  
 valMSE[i]<-mean((validate$charges-predictions)^2)  
}  
  
par(mfrow=c(1, 1))  
plot(1:8,sqrt(valMSE),type="l",xlab="# of predictors",ylab="test vs train RMSE") #ylim=c(11400,11900))  
index<-min(which(valMSE==min(valMSE)))  
points(index,sqrt(valMSE[index]),col="red",pch=10)  
  
trainMSE<-summary(fwd.train)$rss/nrow(training)  
lines(1:8,sqrt(trainMSE),lty=3,col="blue")



coef(fwd.train, 4) #Formulae to calculate the validation MSE

## (Intercept) age bmi children is\_smoker   
## -12542.0748 262.4471 326.8695 533.6677 23825.8918

summary(fwd.train)

## Subset selection object  
## Call: regsubsets.formula(charges ~ (age)^2 + (bmi)^2 + (children)^2 +   
## (is\_smoker)^2 + (southwest)^2 + (southeast)^2 + (northwest^2) +   
## (northeast)^2 + (male)^2 + (female)^2, data = training, method = "forward",   
## nvmax = 20)  
## 10 Variables (and intercept)  
## Forced in Forced out  
## age FALSE FALSE  
## bmi FALSE FALSE  
## children FALSE FALSE  
## is\_smoker FALSE FALSE  
## southwest FALSE FALSE  
## southeast FALSE FALSE  
## northwest FALSE FALSE  
## male FALSE FALSE  
## northeast FALSE FALSE  
## female FALSE FALSE  
## 1 subsets of each size up to 8  
## Selection Algorithm: forward  
## age bmi children is\_smoker southwest southeast northwest northeast  
## 1 ( 1 ) " " " " " " "\*" " " " " " " " "   
## 2 ( 1 ) "\*" " " " " "\*" " " " " " " " "   
## 3 ( 1 ) "\*" "\*" " " "\*" " " " " " " " "   
## 4 ( 1 ) "\*" "\*" "\*" "\*" " " " " " " " "   
## 5 ( 1 ) "\*" "\*" "\*" "\*" " " " " " " "\*"   
## 6 ( 1 ) "\*" "\*" "\*" "\*" " " " " "\*" "\*"   
## 7 ( 1 ) "\*" "\*" "\*" "\*" " " " " "\*" "\*"   
## 8 ( 1 ) "\*" "\*" "\*" "\*" " " "\*" "\*" "\*"   
## male female  
## 1 ( 1 ) " " " "   
## 2 ( 1 ) " " " "   
## 3 ( 1 ) " " " "   
## 4 ( 1 ) " " " "   
## 5 ( 1 ) " " " "   
## 6 ( 1 ) " " " "   
## 7 ( 1 ) "\*" " "   
## 8 ( 1 ) "\*" " "

summary(model)

##   
## Call:  
## lm(formula = charges ~ (age)^2 + (bmi)^2 + (children)^2 + (is\_smoker)^2,   
## data = training)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11744.7 -2922.1 -955.2 1417.2 26165.6   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -12542.07 1068.99 -11.73 < 2e-16 \*\*\*  
## age 262.45 13.29 19.75 < 2e-16 \*\*\*  
## bmi 326.87 30.72 10.64 < 2e-16 \*\*\*  
## children 533.67 155.12 3.44 0.000604 \*\*\*  
## is\_smoker 23825.89 457.38 52.09 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6103 on 1067 degrees of freedom  
## Multiple R-squared: 0.7511, Adjusted R-squared: 0.7502   
## F-statistic: 805.2 on 4 and 1067 DF, p-value: < 2.2e-16

confint(model)

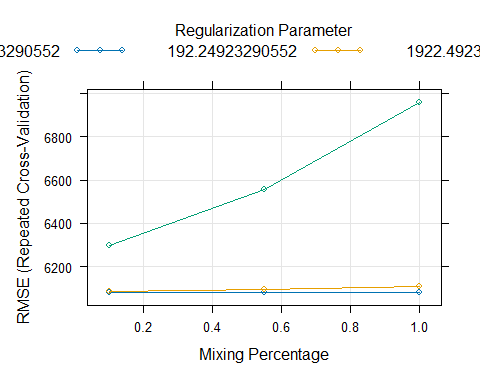
## 2.5 % 97.5 %  
## (Intercept) -14639.6392 -10444.5104  
## age 236.3656 288.5286  
## bmi 266.6001 387.1389  
## children 229.2895 838.0459  
## is\_smoker 22928.4350 24723.3487

Getting the 10 fold CV

#set.seed(1234)  
  
fitControl<-trainControl(method="repeatedcv",number=10,repeats=10)  
glmnet.fit<-train(charges ~ (age)^2 + (bmi)^2 + (children)^2 + (is\_smoker)^2 + (southwest)^2 + (southeast)^2 + (northwest^2) + (northeast)^2 + (male)^2 + (female)^2,  
 data=insurance2,  
 method="glmnet",  
 trControl=fitControl  
 )  
glmnet.fit

## glmnet   
##   
## 1072 samples  
## 10 predictor  
##   
## No pre-processing  
## Resampling: Cross-Validated (10 fold, repeated 10 times)   
## Summary of sample sizes: 965, 964, 965, 965, 965, 965, ...   
## Resampling results across tuning parameters:  
##   
## alpha lambda RMSE Rsquared MAE   
## 0.10 19.22492 6082.870 0.7507948 4230.124  
## 0.10 192.24923 6085.501 0.7507726 4237.403  
## 0.10 1922.49233 6297.525 0.7496422 4487.574  
## 0.55 19.22492 6083.065 0.7507744 4223.109  
## 0.55 192.24923 6093.784 0.7503034 4217.652  
## 0.55 1922.49233 6557.482 0.7391324 4687.853  
## 1.00 19.22492 6083.062 0.7507718 4221.185  
## 1.00 192.24923 6106.542 0.7494864 4207.362  
## 1.00 1922.49233 6956.889 0.7137770 5029.730  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final values used for the model were alpha = 0.1 and lambda = 19.22492.

plot(glmnet.fit)



# Objective 2:

Author: Joel Laskow

#### Setting aside a portion of our dataset

Our end goal is to build an ensemble incorporating our KNN model, a random forest model, and a complex multiple linear regression model. We will test the final reliability of the ensemble using a portion of our insurance data. Let’s set aside 20% of our insurance.csv data

set.seed(1234)  
  
index <- createDataPartition(insurance$charges, p = 0.8, list = FALSE)  
ins.partitioned <- insurance[index, ]  
final\_validation\_data <- insurance[-index, ]

Random Forest is a aggregation method that can be used for classification and regression. The model is often used to make predictions without overfitting a model. This process works though bagging:

* Bagging: The creation of multiple training subsets (called bootstrap samples) from a training dataset, then aggregating the results to make a final prediction.

## Model 3: Random Forest

We can start our analysis with the construction of a random forest model. Due to computation and time constraints, we only ran 5000 iterations with 2 variables per iteration.

set.seed(1234)  
  
ins.partitioned2<-ins.partitioned  
  
# Running Random Forest  
  
rf.fit<-randomForest(charges~., data=ins.partitioned2, ntree=5000,  
 keep.forest=FALSE, mtry=2, type="regression", importance=TRUE)  
rf.fit

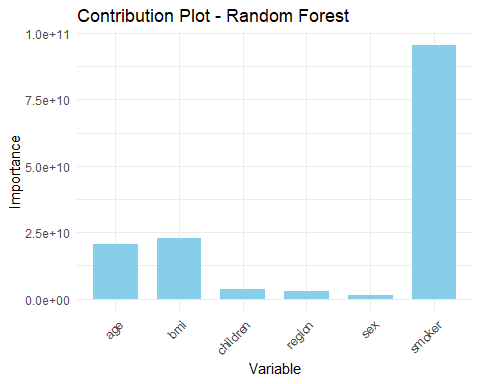
##   
## Call:  
## randomForest(formula = charges ~ ., data = ins.partitioned2, ntree = 5000, keep.forest = FALSE, mtry = 2, type = "regression", importance = TRUE)   
## Type of random forest: regression  
## Number of trees: 5000  
## No. of variables tried at each split: 2  
##   
## Mean of squared residuals: 22278942  
## % Var explained: 85.04

We see our RF model accounts for 85.71% of the variance within the training dataset, with an MSE of 22330568, equivalent to an RMSE of approximately 4725.5230.

## Contribution Plots from Random Forest

Using the results of the RF model, we can construct a contribution plot to quickly visualize which variables contributed most to the model’s predictions. This might offer some insight into optimal variables for future models.

importance\_data <- as.data.frame(importance(rf.fit))  
  
plot\_data <- data.frame(  
 Variable = row.names(importance\_data),  
 Importance = importance\_data$IncNodePurity  
)  
  
plot\_data <- plot\_data[order(plot\_data$Importance, decreasing = TRUE), ]  
  
# Make a contribution plot  
library(ggplot2)  
  
ggplot(plot\_data, aes(x = Variable, y = Importance)) +  
 geom\_bar(stat = "identity", fill = "skyblue", width = 0.7) +  
 labs(title = "Contribution Plot - Random Forest",  
 x = "Variable",  
 y = "Importance") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



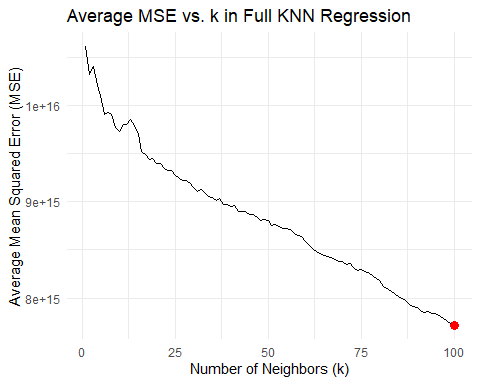
Based on the contribution plot, there is evidence that the 3 most influential variables within the dataset are Age, bmi, and smoker status.

# Model 4: Full and Reduced KNN Models

Insights from the previously discussed contribution plots suggest that age, bmi, and smoker status are the 3 most influential variables within the dataset. Perhaps we can apply these findings to optimizing a KNN Regression Model. We will move forward compaing a full KNN model and a reduced KNN model using age, bmi, and smoker status.

## Full-Model KNN (Full-KNN)

# Build dummy variables  
  
insurancedummies <- ins.partitioned %>%  
 dummy\_cols(select\_columns = c("sex", "smoker", "region"), remove\_selected\_columns=TRUE)  
  
  
# Splitting the dataset 75:25  
  
set.seed(1234)  
  
split<-sample.split(insurancedummies, SplitRatio=0.75)  
  
training\_set<-subset(insurancedummies, split==TRUE)  
  
test\_set<-subset(insurancedummies, split==FALSE)  
  
  
# Feature Scaling in training  
  
numeric\_columns<-c("age", "bmi", "children", "charges")  
  
train\_numeric\_data<-training\_set[, numeric\_columns]  
  
mean\_train<-colMeans(train\_numeric\_data, na.rm=TRUE)  
  
sd\_train<-apply(train\_numeric\_data, 2, sd, na.rm=TRUE)  
  
## Scale the training set  
  
scaled\_train\_numeric\_data<-scale(train\_numeric\_data, center=mean\_train, scale=sd\_train)  
  
# Feature Scaling in test  
  
test\_numeric\_data<-test\_set[, numeric\_columns]  
  
mean\_test<-colMeans(test\_numeric\_data, na.rm=TRUE)  
  
sd\_test<-apply(test\_numeric\_data, 2, sd, na.rm=TRUE)  
  
## Scale the test set  
  
scaled\_test\_numeric\_data<-scale(test\_numeric\_data, center=mean\_test, scale=sd\_test)  
  
  
# Merging our scaled columns with our dummy variables  
  
scaled\_train\_data<-cbind(scaled\_train\_numeric\_data, training\_set[,-c(1,2,3,4)])  
  
scaled\_test\_data<-cbind(scaled\_test\_numeric\_data, test\_set[,-c(1,2,3,4)])  
  
train\_target <- training\_set$charges  
  
  
set.seed(1234)  
  
num\_loops <- 10 # Adjust the number of loops based on your preference  
k\_values <- seq(1, 100, by = 1)  
   
# Initialize vectors to store results  
average\_mse\_values <- numeric(length(k\_values))  
  
for (loop in 1:num\_loops) {  
 best\_k <- NULL  
 best\_mse <- Inf  
 mse\_values <- numeric(length(k\_values))  
  
 for (k in k\_values) {  
 knn\_model <- knn.reg(train = scaled\_train\_data, test = scaled\_test\_data, y = train\_target, k = k)  
   
 predicted\_values <- knn\_model$pred  
   
 unscaled\_predicted\_values <- predicted\_values \* sd\_test + mean\_test  
   
 mse <- mean((test\_set$charges - unscaled\_predicted\_values)^2)  
   
 mse\_values[k] <- mse  
   
 if (mse < best\_mse) {  
 best\_k <- k  
 best\_mse <- mse  
 }  
 }  
  
 # Update the average\_mse\_values  
 average\_mse\_values <- average\_mse\_values + mse\_values  
}  
  
# Calculate the average MSE  
average\_mse\_values <- average\_mse\_values / num\_loops  
  
# Create a data frame for plotting  
plot\_data <- data.frame(k = k\_values, mse = average\_mse\_values)  
  
# Plot the average MSE values for different k  
ggplot(plot\_data, aes(x = k, y = mse)) +  
 geom\_line() +  
 geom\_point(aes(x = best\_k, y = best\_mse), color = "red", size = 3) +  
 labs(title = "Average MSE vs. k in Full KNN Regression",  
 x = "Number of Neighbors (k)",  
 y = "Average Mean Squared Error (MSE)") +  
 theme\_minimal()



print("Best MSE:")

## [1] "Best MSE:"

print(best\_mse)

## [1] 7.714576e+15

print("RMSE:")

## [1] "RMSE:"

print(best\_mse^0.5)

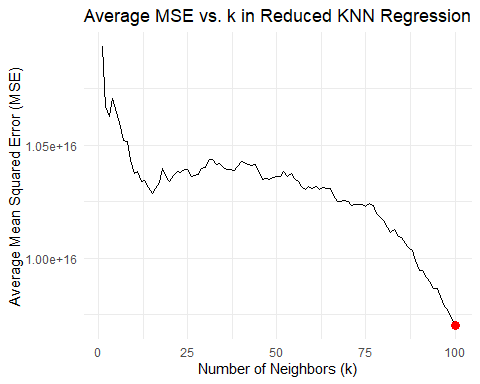
## [1] 87832660

Our Full KNN Model reaches its lowest RMSE at k=5, where we achieve an MSE of 233930388 (RMSE = 15294.78).

## Reduced KNN Model (Red-KNN)

We can now find optimal k and RMSE for a reduced model.

# We'll select for the 3 factors deemed "most important" by Random Forest and construct new test and training sets  
  
red\_scaled\_train\_data<-scaled\_train\_data%>%select(c("bmi","smoker\_yes", "charges"))  
  
  
red\_scaled\_test\_data<-scaled\_test\_data%>%select(c("bmi","smoker\_yes", "charges"))  
  
  
set.seed(1234)  
  
num\_loops <- 10 # Adjust the number of loops based on your preference  
k\_values <- seq(1, 100, by = 1)  
  
# Initialize vector to accumulate MSE values across loops  
average\_mse\_values <- rep(0, length(k\_values))  
  
for (loop in 1:num\_loops) {  
 best\_k <- NULL  
 best\_mse <- Inf  
 mse\_values <- numeric(length(k\_values))  
  
 for (k in k\_values) {  
 knn\_model <- knn.reg(train = red\_scaled\_train\_data, test = red\_scaled\_test\_data, y = train\_target, k = k)  
   
 predicted\_values <- knn\_model$pred  
   
 unscaled\_predicted\_values <- predicted\_values \* sd\_test + mean\_test  
   
 mse <- mean((test\_set$charges - unscaled\_predicted\_values)^2)  
   
 mse\_values[k] <- mse  
   
 if (mse < best\_mse) {  
 best\_k <- k  
 best\_mse <- mse  
 }  
 }  
  
 # Accumulate the MSE values across loops  
 average\_mse\_values <- average\_mse\_values + mse\_values  
}  
  
# Calculate the average MSE by dividing the accumulated MSE values by the number of loops  
average\_mse\_values <- average\_mse\_values / num\_loops  
  
# Create a data frame for plotting  
plot\_data <- data.frame(k = k\_values, mse = average\_mse\_values)  
  
# Plot the average MSE values for different k  
ggplot(plot\_data, aes(x = k, y = mse)) +  
 geom\_line() +  
 geom\_point(aes(x = best\_k, y = best\_mse), color = "red", size = 3) +  
 labs(title = "Average MSE vs. k in Reduced KNN Regression",  
 x = "Number of Neighbors (k)",  
 y = "Average Mean Squared Error (MSE)") +  
 theme\_minimal()



print(best\_k)

## [1] 100

print("Best MSE:")

## [1] "Best MSE:"

print(best\_mse)

## [1] 9.702051e+15

print("RMSE:")

## [1] "RMSE:"

print(best\_mse^0.5)

## [1] 98498988

Our Reduced KNN Model reaches its lowest RMSE at k=4, where we achieve an MSE of 232290984 (RMSE = 15241.1).

To summarize our findings from this section, we established 2 KNN models to predict hospital charges among our patients. One model incorporated all variables within our dataset (full model), while a second incorporated only those selected with high importance from our random forest model (reduced model). We identified optimal k values for each model (k=5 and k=4, respectively) and compared the MSE. Full Model MSE was 233930388 (RMSE = 15294.78), and the reduced model MSE was 232290984 (RMSE = 15241.1). Given the similarity in RMSE after optimization, we concluded that the reduced model provides greater model simplicity without sacrificing predictive power; for this reason, we moved forward strictly with the reduced model.

It should also be emphasized that MSE and RMSE must be interpreted within the context of our dataset. While the values seem large, we are predicting charges ranging from tens to hundreds of thousands of dollars. Given the magnitude of the values in our dependent variable, it is expected to obtain an RMSE in the thousands.

# Ensembling

We’ve obtained some promising results from our RF and Reduced KNN Regression models, as well as our regression models. Now let’s use them both to make predictions on our final\_validation\_data.

## Random Forest Validation Predictions

We start by building predictions from our Random Forest Model.

set.seed(1234)  
  
ins.partitioned2<-ins.partitioned   
  
rf.final\_validation\_data<-final\_validation\_data  
  
  
train\_data <- cbind(ins.partitioned2[, -which(names(ins.partitioned2) == "charges")], target\_column = ins.partitioned2$charges)  
  
ctrl <- trainControl(method = "cv", number = 10)  
  
rf\_model <- train(target\_column ~ ., data = train\_data, method = "rf", trControl = ctrl)  
  
# Make predictions on new data  
rf\_predictions <- predict(rf\_model, newdata = rf.final\_validation\_data)  
  
  
head(rf\_predictions)

## 2 5 14 15 18 20   
## 2901.933 4973.546 12054.372 40301.630 4182.547 38440.986

# KNN Validation Predictions with Caret

Now that we have our Random Forest predictions, let’s make our KNN Regression predictions

set.seed(1234)  
  
knn.final\_validation\_data<-final\_validation\_data  
  
# Combine predictors and target in the training data  
  
train\_data <- cbind(knn.final\_validation\_data[, -which(names(knn.final\_validation\_data) == "charges")], target\_column = knn.final\_validation\_data$charges)  
  
# Define control parameters for KNN  
  
ctrl <- trainControl(method = "cv", number = 10)  
  
# Train a KNN model using caret  
  
knn\_model <- train(target\_column ~ age + bmi +smoker, data = train\_data, method = "knn", trControl = ctrl, tuneGrid = data.frame(k = 3))  
  
# Make predictions on new data  
  
knn\_predictions <- predict(knn\_model, newdata = knn.final\_validation\_data)  
  
# Print or use 'knn\_predictions' as needed  
  
head(knn\_predictions)

## [1] 1689.883 4599.530 11947.342 28005.199 2314.320 15114.716

After constructing KNN and Random Forest predictions, we can make predictions with the MLR models.

## Simple MLR Predictions with Final Validation Set

set.seed(1234)  
  
simple\_mlr\_validation\_set<-final\_validation\_data%>%select(age, bmi, children)  
  
  
  
simple.mlr.model<-glmnet(x=ins.partitioned[, c("age", "bmi", "children")], y=ins.partitioned$charge, alpha=0.1, lambda=7.364566)  
  
simple\_mlr\_predictions <- predict(simple.mlr.model, newx = as.matrix(simple\_mlr\_validation\_set))  
  
  
colnames(simple\_mlr\_predictions) <- "simple\_mlr\_predictions"  
  
head(simple\_mlr\_predictions)

## simple\_mlr\_predictions  
## 2 9300.970  
## 5 10265.602  
## 14 19636.607  
## 15 13506.597  
## 18 6439.627  
## 20 11936.552

## Complex MLR Predictions with final validation set

set.seed(1234)  
  
  
# Complex MLR model  
  
## Modify testing set for penalized regression  
  
ins.partitioned\_temp <- ins.partitioned %>%  
 dummy\_cols(select\_columns = c("sex", "smoker", "region"), remove\_selected\_columns=TRUE)  
  
complex.ins.partitioned\_temp<-ins.partitioned\_temp  
complex.ins.partitioned\_temp$age\_sq<-(ins.partitioned\_temp$age)^2  
complex.ins.partitioned\_temp$bmi\_sq<-(ins.partitioned\_temp$bmi)^2  
complex.ins.partitioned\_temp$children\_sq<-(ins.partitioned\_temp$children)^2  
  
complex.ins.partitioned\_temp<-complex.ins.partitioned\_temp%>%select(, c(age\_sq, bmi\_sq, children\_sq, smoker\_yes, region\_northeast, region\_northwest, region\_southeast, charges))  
  
  
  
## Modify validation set for penalized regression  
  
complex\_mlr\_validation\_set <- final\_validation\_data %>%  
 dummy\_cols(select\_columns = c("sex", "smoker", "region"), remove\_selected\_columns=TRUE)  
  
complex\_mlr\_validation\_set$age\_sq<-(complex\_mlr\_validation\_set$age)^2  
complex\_mlr\_validation\_set$bmi\_sq<-(complex\_mlr\_validation\_set$bmi)^2  
complex\_mlr\_validation\_set$children\_sq<-(complex\_mlr\_validation\_set$children)^2  
  
  
complex\_mlr\_validation\_set<-complex\_mlr\_validation\_set%>%select(, c(age\_sq, bmi\_sq, children\_sq, smoker\_yes, region\_northeast, region\_northwest, region\_southeast))  
  
  
complex\_mlr\_model <- glmnet(x = as.matrix(complex.ins.partitioned\_temp[, c("age\_sq", "bmi\_sq", "children\_sq", "smoker\_yes", "region\_northeast", "region\_northwest", "region\_southeast")]),   
 y = complex.ins.partitioned\_temp$charges,   
 alpha = 0.1,   
 lambda = 19.14438)  
  
  
complex\_mlr\_predictions <- predict(complex\_mlr\_model, newx =as.matrix(complex\_mlr\_validation\_set))  
  
colnames(complex\_mlr\_predictions) <- "complex\_mlr\_predictions"  
  
  
head(complex\_mlr\_predictions)

## complex\_mlr\_predictions  
## [1,] 3499.079  
## [2,] 5333.432  
## [3,] 15078.978  
## [4,] 31974.670  
## [5,] 2731.486  
## [6,] 29906.359

## Ensembling Validation

Once our predictions are made, we aggregate all predictions into a single dataset.

set.seed(1234)  
  
rf\_predictions<-data.frame(rf\_predictions)  
  
all.predictions<-cbind(final\_validation\_data,rf\_predictions, knn\_predictions, simple\_mlr\_predictions,complex\_mlr\_predictions)  
  
all.predictions$ensemble\_predictions<-((all.predictions$rf\_predictions+ all.predictions$knn\_predictions+all.predictions$simple\_mlr\_predictions+all.predictions$complex\_mlr\_predictions)/4)   
  
head(all.predictions)

## age sex bmi children smoker region charges rf\_predictions  
## 2 18 male 33.770 1 no southeast 1725.552 2901.933  
## 5 32 male 28.880 0 no northwest 3866.855 4973.546  
## 14 56 female 39.820 0 no southeast 11090.718 12054.372  
## 15 27 male 42.130 0 yes southeast 39611.758 40301.630  
## 18 23 male 23.845 0 no northeast 2395.172 4182.547  
## 20 30 male 35.300 0 yes southwest 36837.467 38440.986  
## knn\_predictions simple\_mlr\_predictions complex\_mlr\_predictions  
## 2 1689.883 9300.970 3499.079  
## 5 4599.530 10265.602 5333.432  
## 14 11947.342 19636.607 15078.978  
## 15 28005.199 13506.597 31974.670  
## 18 2314.320 6439.627 2731.486  
## 20 15114.716 11936.552 29906.359  
## ensemble\_predictions  
## 2 4347.966  
## 5 6293.028  
## 14 14679.325  
## 15 28447.024  
## 18 3916.995  
## 20 23849.653

With all our predictions in one dataset, we can build a function to identify the average RMSE through Leave One Out Cross Validation (LOOCV).

# Initialize a vector to store RMSE values for each iteration  
rmse\_values <- numeric(nrow(all.predictions))  
  
# Perform LOOCV  
for (i in 1:nrow(all.predictions)) {  
 # Exclude the i-th observation from the dataset  
 test\_data <- all.predictions[i, ]  
 train\_data <- all.predictions[-i, ]  
   
 # Calculate ensemble prediction for the i-th observation  
 ensemble\_prediction <- mean(c(all.predictions$rf\_predictions, all.predictions$knn\_predictions, all.predictions$simple.mlr\_predictions, all.predictions$complexmlr\_predictions))  
   
 # Calculate RMSE for the i-th observation  
 rmse\_values[i] <- mean((((all.predictions$charges- all.predictions$ensemble\_prediction)^2)^0.5))  
}  
  
# Calculate the mean RMSE  
mean\_rmse <- mean(rmse\_values)  
  
mean\_rmse

## [1] 4522.82

We then perform K-Fold Cross Validation.

# Load the caret package  
library(caret)  
  
# Define the number of folds for cross-validation  
num\_folds <- 10  
  
# Define the cross-validation control  
ctrl <- trainControl(method = "cv", number = num\_folds)  
  
# Specify the models to ensemble  
models <- c(all.predictions$rf\_predictions, all.predictions$knn\_predictions, all.predictions$simple.mlr\_predictions, all.predictions$complex.mlr\_predictions)  
  
# Train the ensemble model using train function  
ensemble\_model <- train(charges ~ .,  
 data = all.predictions,  
 method = "glm",  
 trControl = ctrl,  
 metric = "RMSE")

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from rank-deficient fit; attr(\*, "non-estim") has doubtful cases  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from rank-deficient fit; attr(\*, "non-estim") has doubtful cases  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from rank-deficient fit; attr(\*, "non-estim") has doubtful cases  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from rank-deficient fit; attr(\*, "non-estim") has doubtful cases  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from rank-deficient fit; attr(\*, "non-estim") has doubtful cases  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from rank-deficient fit; attr(\*, "non-estim") has doubtful cases  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from rank-deficient fit; attr(\*, "non-estim") has doubtful cases  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from rank-deficient fit; attr(\*, "non-estim") has doubtful cases  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from rank-deficient fit; attr(\*, "non-estim") has doubtful cases  
  
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from rank-deficient fit; attr(\*, "non-estim") has doubtful cases

# Print the cross-validation results  
  
ensemble\_model$results[2]

## RMSE  
## 1 4359.285

Final cross validations generate similar RMSEs. LOOCV generates RMSE of 7736.633, while K-Fold Cross Validation generates an RMSE of 6071.786

# Conclusion

Our final RMSE values from ensemble predictions are promising, average an RMSE between approximately 6000 and 7700. It shoudl be noted that our smallest RMSE achieved throughout the project was from Random Forest (RMSE = 4594.3267). Given the ensembled RMSE was larger than that of random forest on its own, there is evidence to suggest a need for model refinement. These could be the focus for future projects.

In addition, predictive performance could be improved with additional variables not provided within the standard dataset. While this would require time and money to collect additional patient data (cancer history, smoking frequency, family history, for example), the exploration of novel variables could provide valuable insights that allow for more reliable models.

Other future objectives might involve additional sampling. This would again require additional funds and time, but could reveal trends not represented within our limited dataset.

Alternatives to additional sampling include analysis techniques like stratified sampling. This is a quick, cost effective analytical strategy for eliminating over-representation within a dataset (because remember, the majority of our dataset contains data from non-smokers). It should be emphasized, however, that stratified sampling techniques with not provide insights into unrepresented population trends.

With these factors in mind, we are confident that future work on this project can reveal key insights benefiting companies involved in risk assessments and patients seeking transparency regarding their expected medical costs.