STAT 6214(Prof. Barut)

Ye In Jeon

**Final Project**

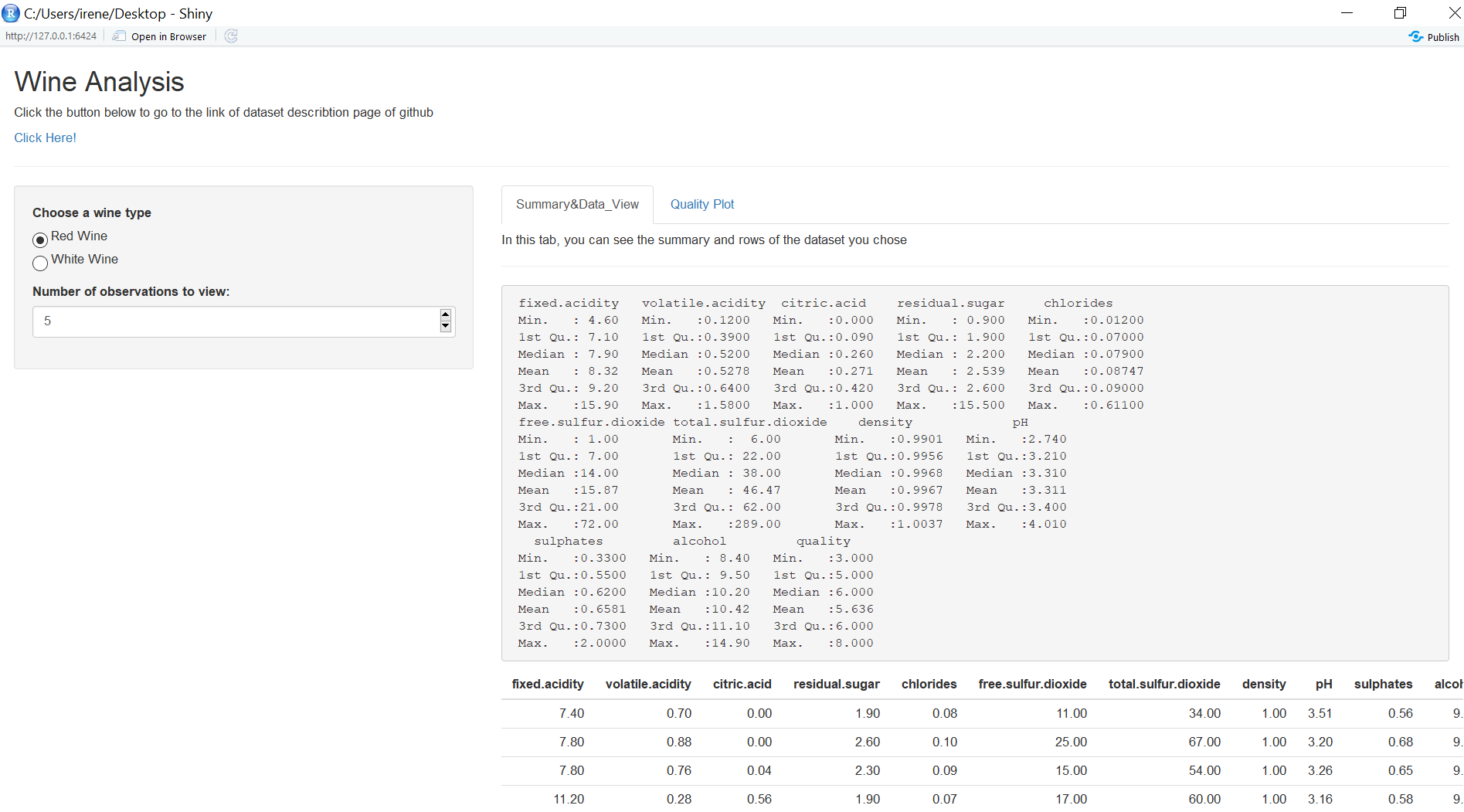
Wine Analysis

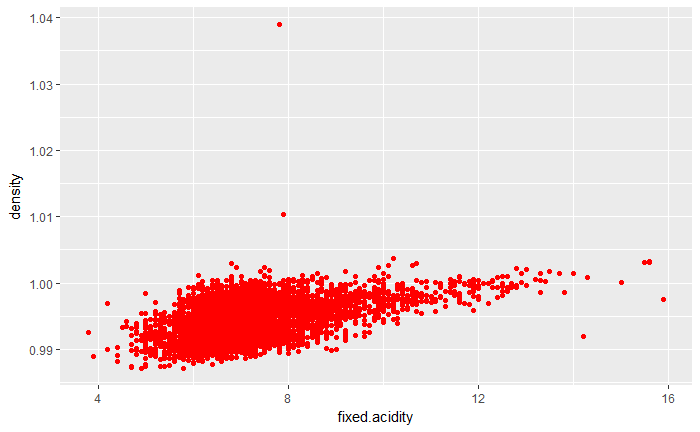
Introduction

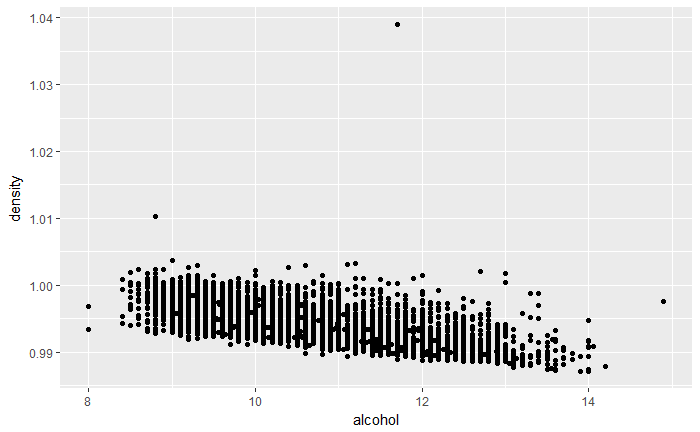
In this project, it contains the wine quality prediction with the chemical elements. In this analysis, the datasets are contained from Kaggle (<https://www.kaggle.com/maitree/wine-quality-selection>) which is divided into red wine and white wine. Both datasets have following columns: Fixed acidity, Volatile acidity, Citric acid, Residual sugar, Chlorides, Free sulfur dioxide, Total sulfur dioxide, Density, pH, Sulphates, Alcohol and Quality. For better analysis, two datasets are merged to one dataset. For the prediction, linear models are built. First, the summary of the dataset and the exploratory data analysis is proceeded. Moreover, several models are built in different methods. Furthermore, the models are evaluated by the cross validation Mean Squared Error (which is called as MSE) value to find the best model. For more information, various packages including ‘e1071’, ‘corrplot’, ‘ggplot2’, ‘rsconnect’, ‘shiny’, ‘dplyr’, ‘MASS’, ‘boot’, ‘glmnet’, ‘car’, ‘gamclass’, ‘mgcv’ and ‘huge’ are used for this analysis.

Exploratory Data Analysis

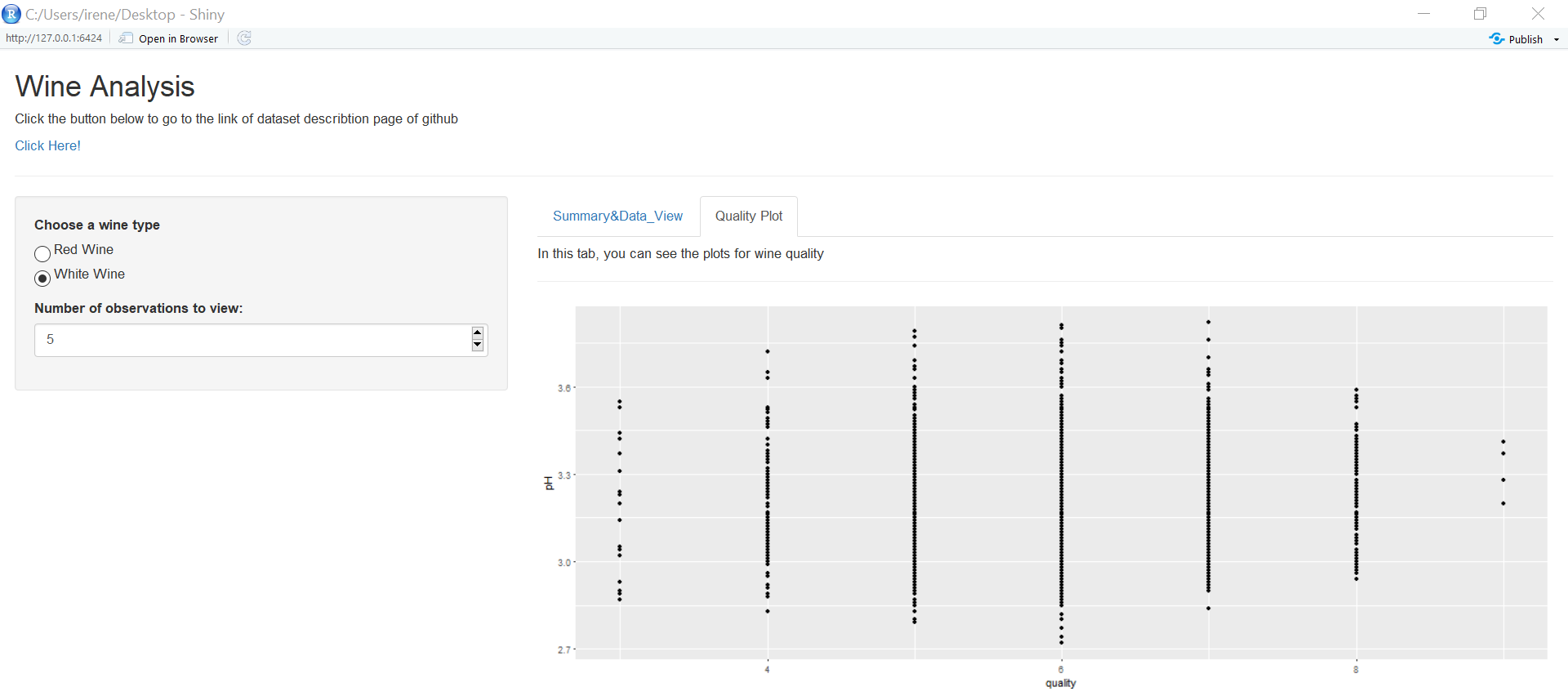
Before starting the analysis, it is important to understand the dataset.



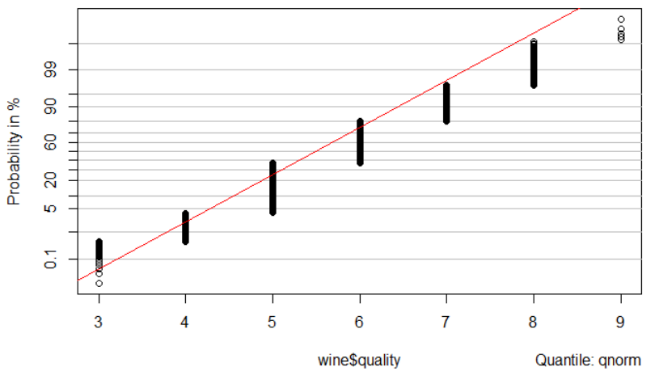
Moreover, the data visualization is progressed. First, as it can be found in the scatter plot below, the positive relationship between density and fixed acidity.

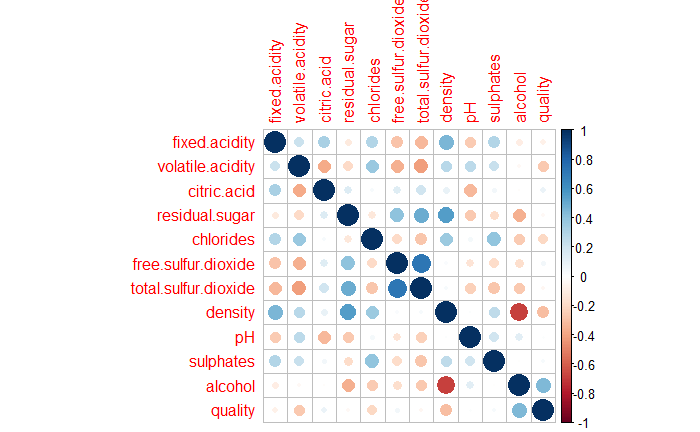
Moreover, the negative relationship between density and alcohol in the plot below.

Furthermore, the scatter plot of quality and pH is proceeded.

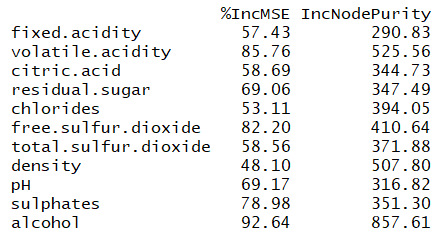


It is important to find out more about the target variable, ‘quality’. According to the analysis, skewness is 0.1895351 which can be said slightly skewed. Moreover, with the probability plot of ‘quality’ variable, it is merely skewed.



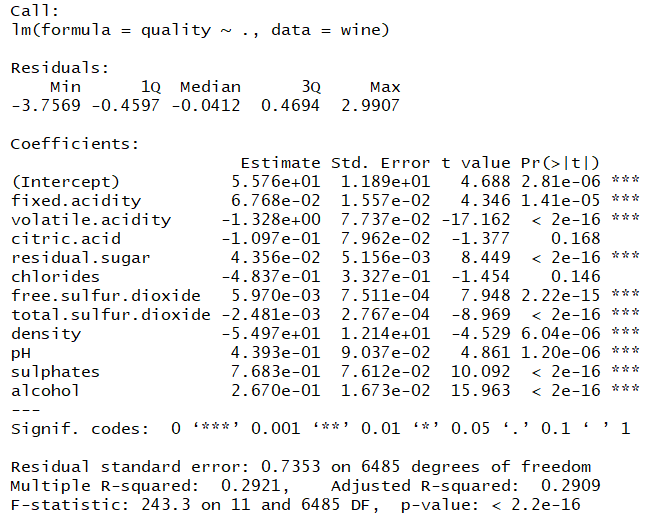
To find the relevance between the features, the correlation test is continued.

To be more specific, random forest feature importance value of each feature is found. According to the values below, it can be found the most values are over 50%. Density feature is 48.10% but it can be rounded to 50%.

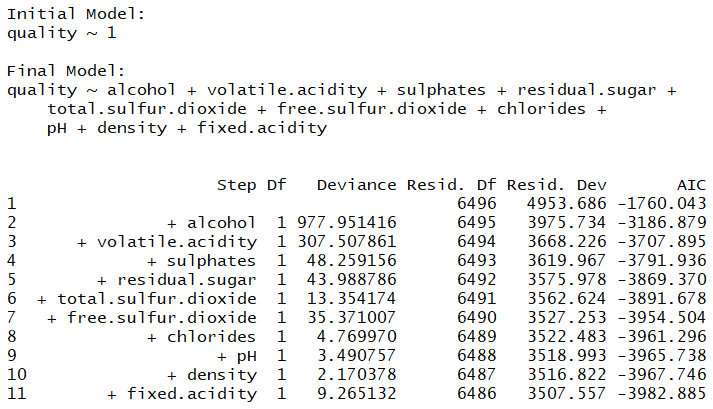


Model building

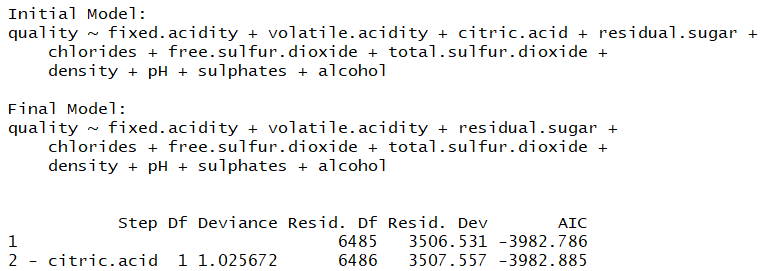
Since, most values of feature importance are around or above 50%, the initial model is built with all the variables in the dataset except the target variable.



The cross validation MSE value of full model is 0.5423346. However, according to the result above, it states that some variables are not statistically significant. Therefore, both backward and forward stepwise feature selection are proceeded.

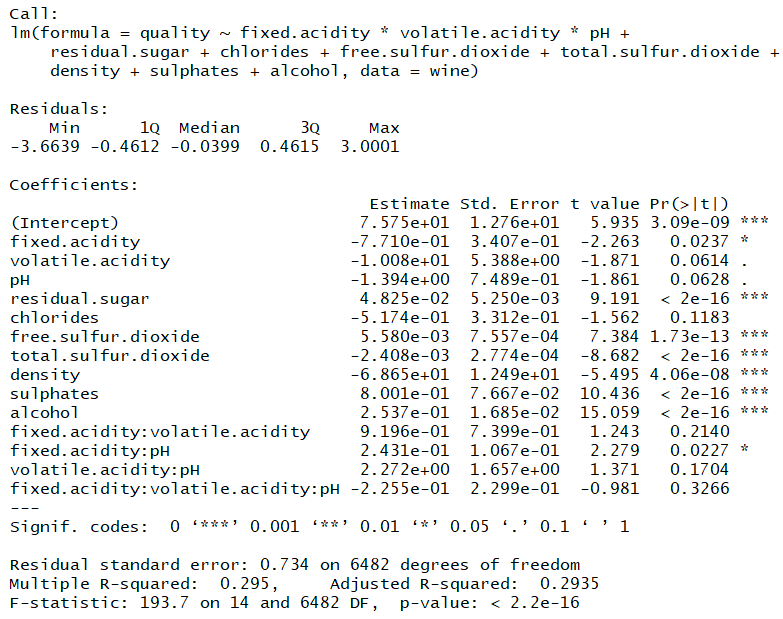


With the forward stepwise selection, the final model includes most of the variables in the dataset except ‘citric.acid’ variable.

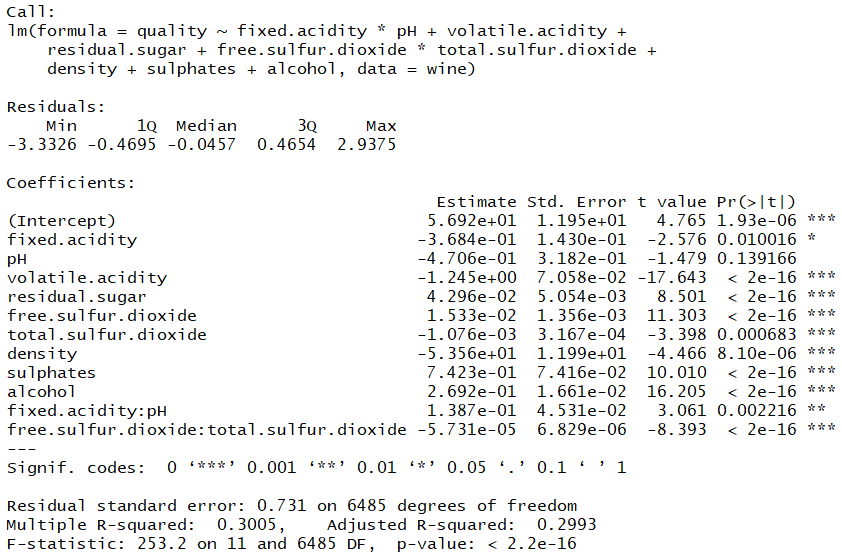


Likewise, the backward stepwise selection returns the same model as the forward stepwise selection. The cross validation MSE value of stepwise model is 0.5421045 which is lower than the full model.

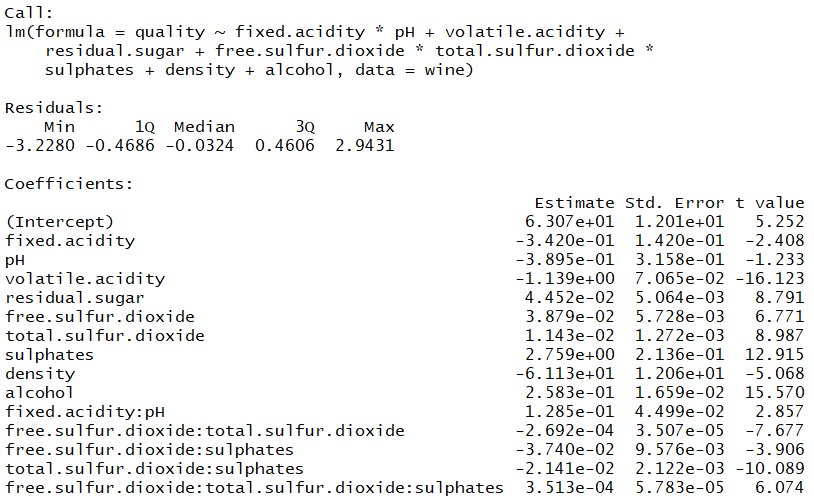
To reduce the cross validation MSE value, interaction terms are added to the model. Since acidity in related to pH, ‘fixed.acidity’, ‘volatile.acidity’ and ‘pH’ are selected to be as the interaction term.

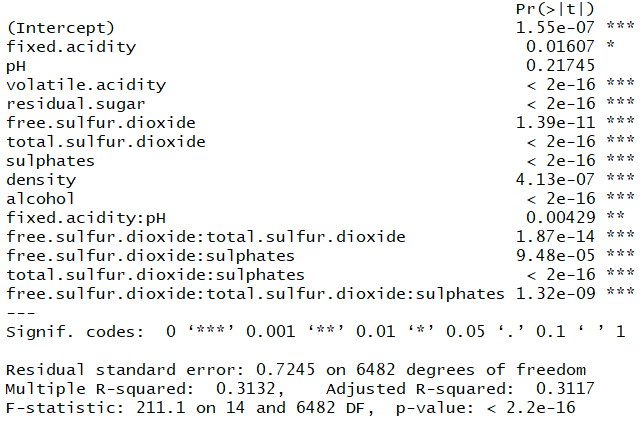


The cross validation MSE value is 0.5407963 which is lower than stepwise model. To reduce this value, another regression model with interaction is tried.



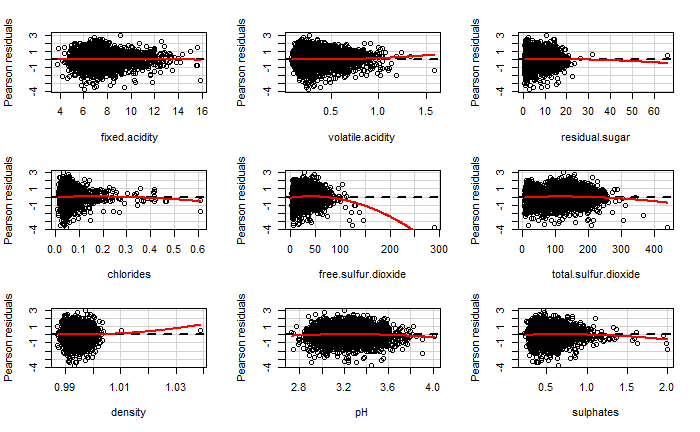
The cross validation MSE value is 0. 5358965 which is lower than the previous interaction term model. To improve the result, ‘free.sulfur.dioxide’, ‘total.sulfur.dioxide’ and ‘sulphates’ are added as another interaction term.

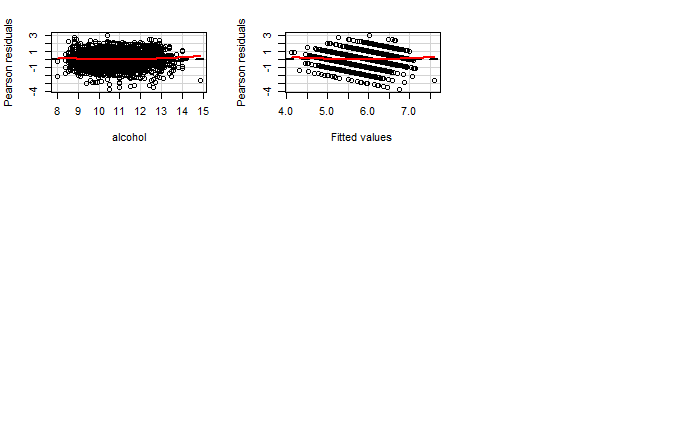




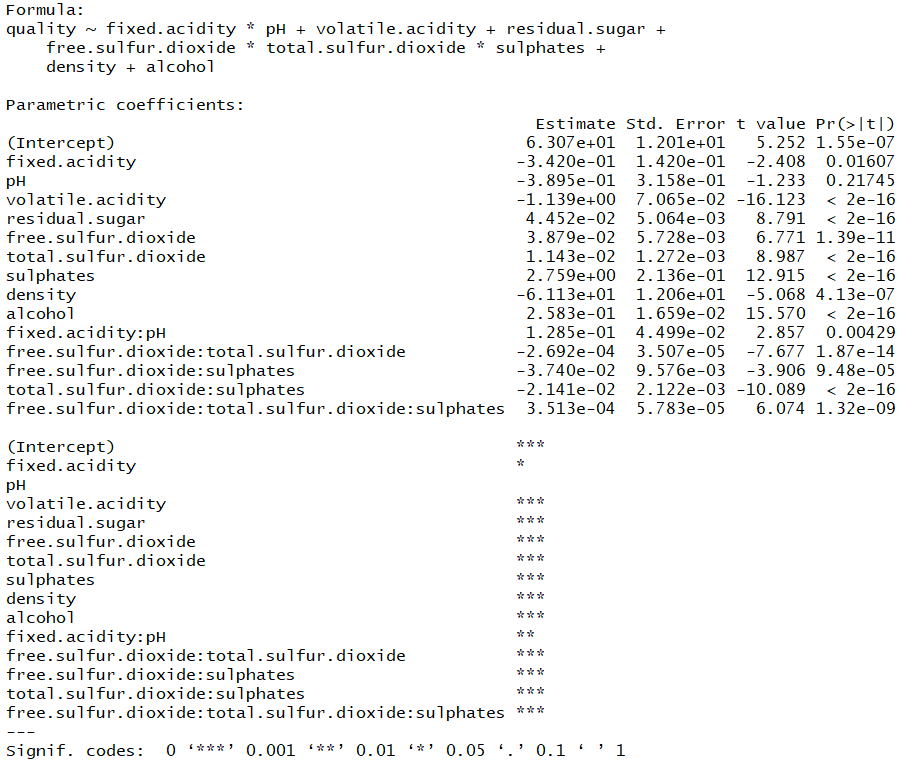
The cross validation MSE value is 0.5268377 which is the lowest value of all the models.

To analyze the linearity of each feature, the residual plots are drawn.





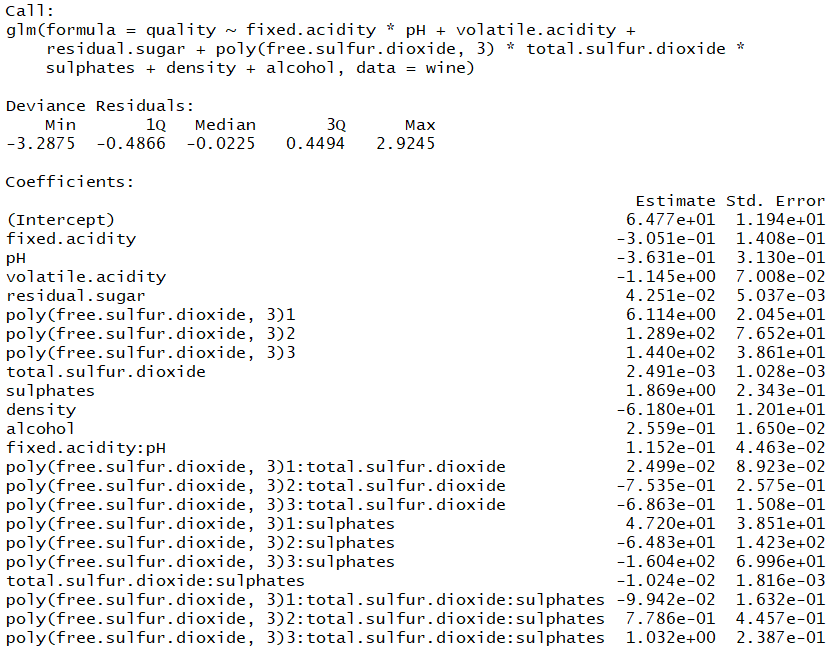
With the plots above, the linearity of ‘free.sulfur.dioxide’ is not proven. Therefore, the spline can be added with this variable.

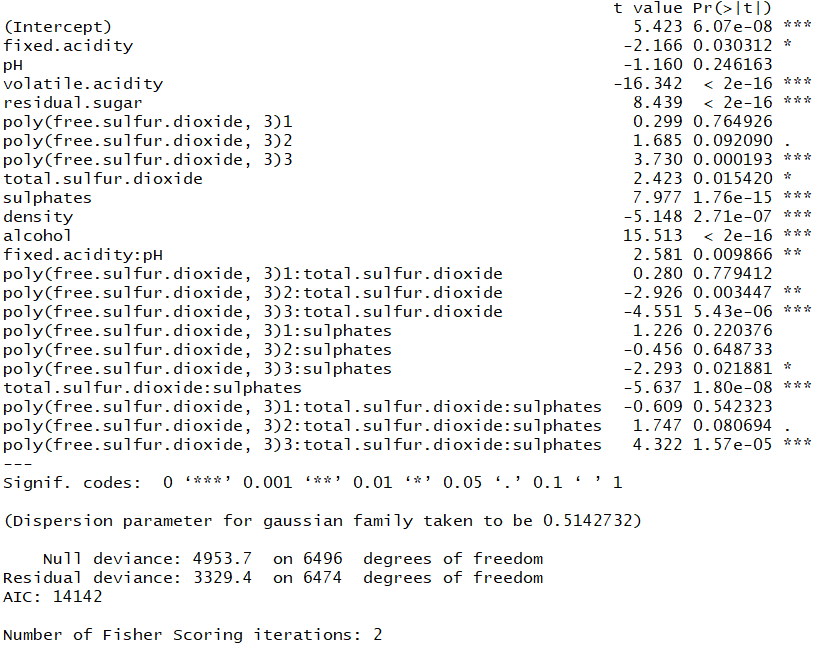




The cross validation MSE value is 0.52743 which is higher than the previous model.

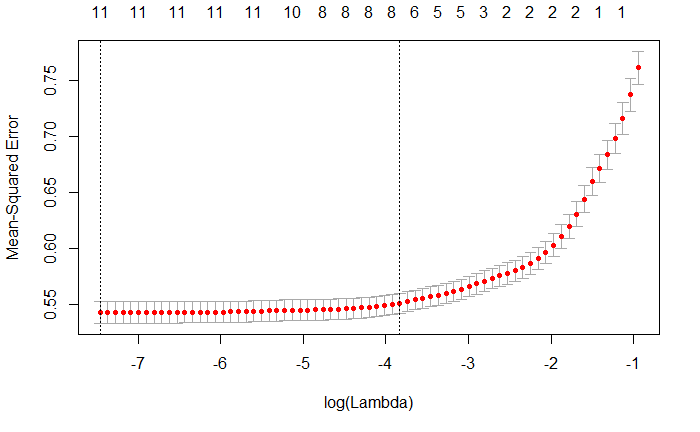
The transformation of ‘free.sulfur.dioxide’ is proceeded.



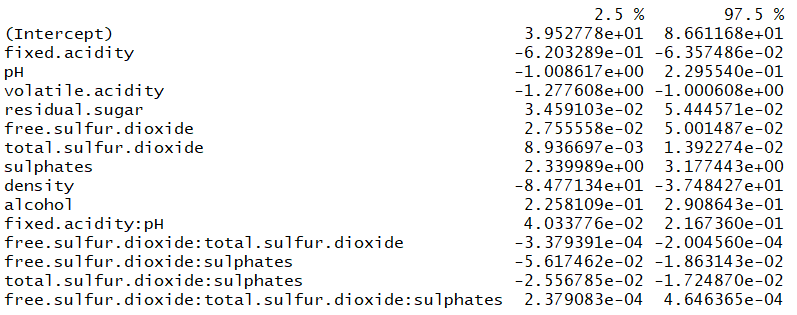


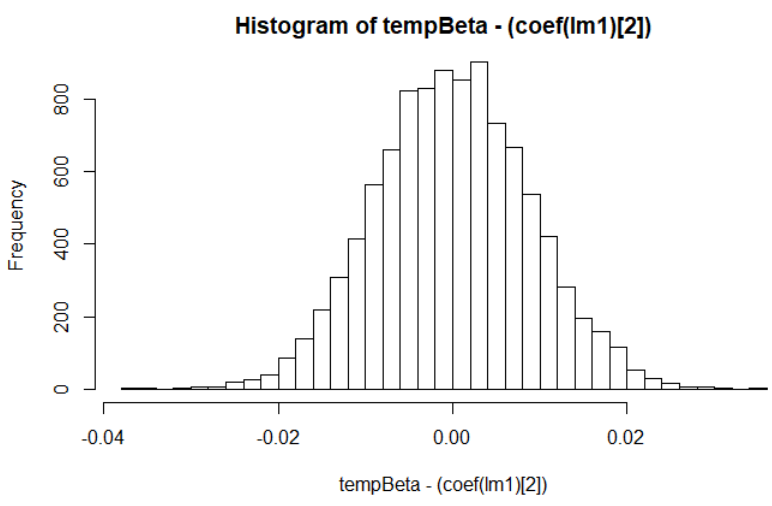
0.8267407 is the value of cross validation MSE which is much higher than the value in the model with two interaction terms.

For better analysis, Lasso regression is proceeded. The result returns the minimum lambda value is 0.0005761285 and the plot as below. The cross validation MSE value of Lasso model is 0.5419758 which is not appropriate.

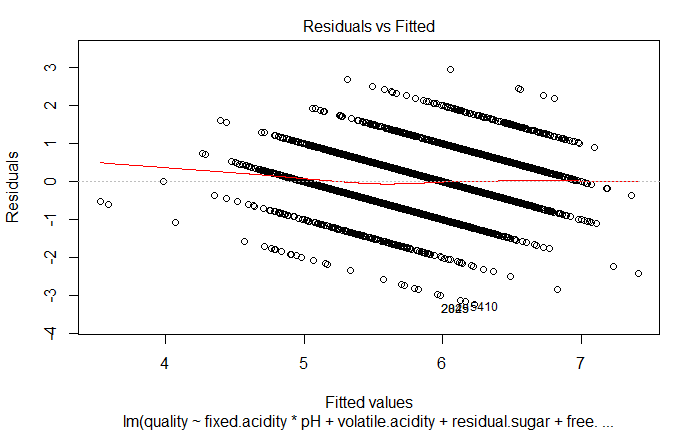
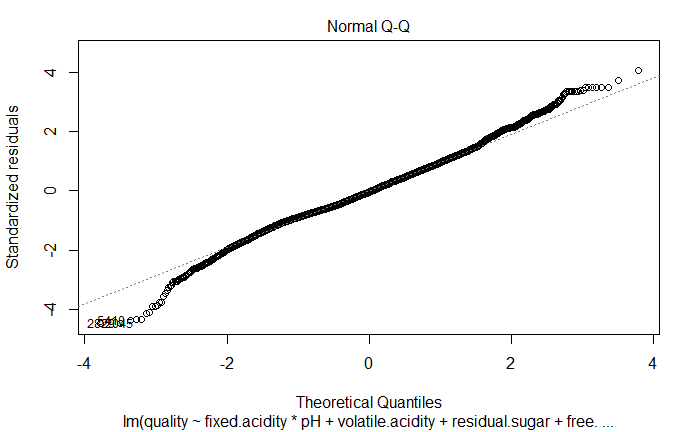


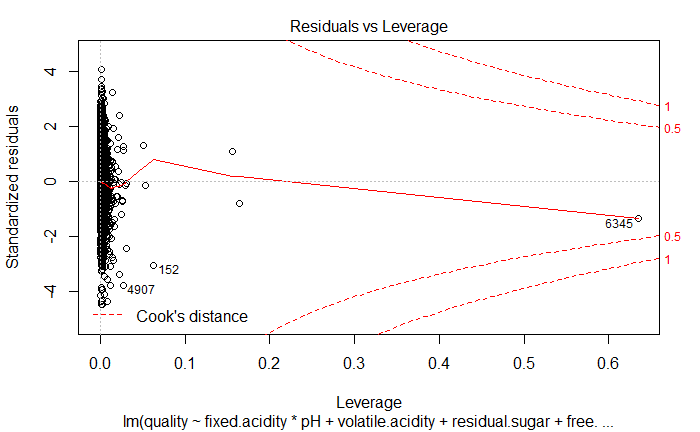
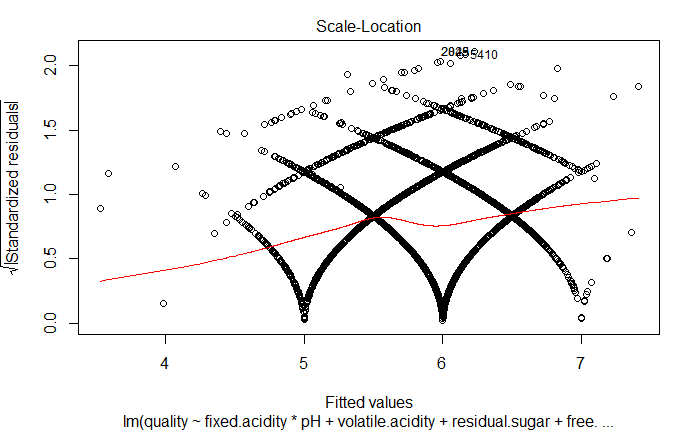
Therefore, the best model is the model with two interaction terms and excluding ‘citric.acid’. First interaction term is ‘free.sulfur.dioxide’, ‘total.sulfur.dioxide’ and ‘sulphates’. In addition, the other interaction term is ‘fixed.acidity’ and ‘pH’. The following is the confidence intervals and the histogram of the model coefficients.





The normality and the independence assumptions can be proven in the following plots:

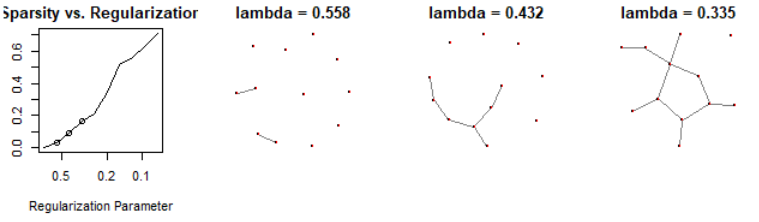




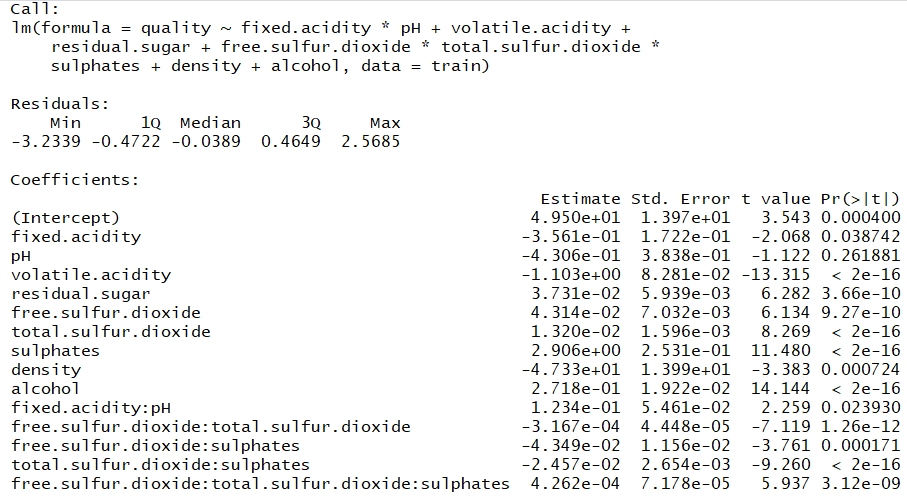
For model diagnosis, the comparison of confidence intervals between normal approximation and bootstrap.

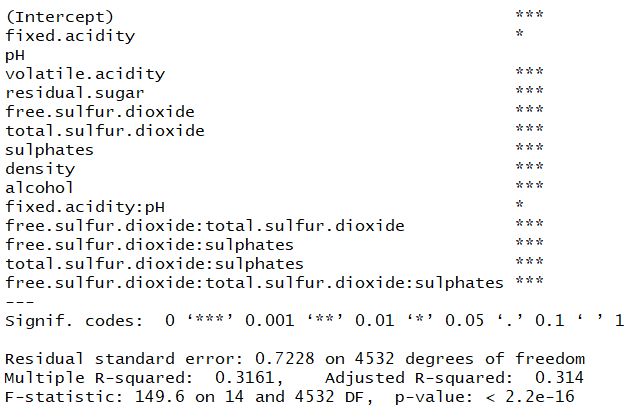
Confidence interval is a measure of significance for regression coefficients. As it can be seen above, the bootstrap confidence interval is minimal difference between normal approximation and bootstrap to some extent.

With ‘huge’ package, the different plots can be drawn with each lambda values. As it can be found below: as the lambda value increases, the more connection between lines are created.



Conclusion

 To see the model application, the dataset is split to train and test dataset in 70% and 30% ratio. When the model is applied to the train dataset, the result is as following:



Next, the test model is predicted with the same model. As it can be seen below: the adjusted R squared value of the train dataset and R squared value of the test dataset is very similar.



Appendix

```{r data, echo=FALSE}

#read dataset

red <- read.csv("D:/R/winequality-red.csv")

white <- read.csv("D:/R/winequality-white.csv")

#merge dataset redwine=1 whitewine=2

wine <- rbind(red, white)

#data structure

str(wine)

summary(wine)

```

```{r skew}

install.packages("e1071")

library(e1071)

skewness(wine$quality, na.rm = FALSE)

probplot(wine$quality, qdist=qnorm, probs=NULL, line=TRUE,

xlab=NULL, ylab="Probability in %")

```

## Importance

```{r randomforest}

wine.rf <- randomForest(quality ~ ., data=wine, importance=TRUE,proximity=TRUE)

round(importance(wine.rf), 2)

```

## Correlation

```{r cor, echo=FALSE}

library(corrplot)

cor <- cor(wine)

corrplot(cor)

```

## EDA Plots

```{r plot1, echo=FALSE}

library(ggplot2)

#scatter plot

ggplot(data=wine, aes(x=fixed.acidity, y=density))+geom\_point(color="red")

```

```{r plot2, echo=FALSE}

#boxplot

library(ggplot2)

ggplot(data=wine, aes(x=alcohol, y=density))+geom\_point()

```

```{r shiny}

install.packages("rsconnect")

library(rsconnect)

library(shiny)

# UI

ui <- fluidPage(

# Title

titlePanel("Wine Analysis"),

tags$h5("Click the button below to go to the link of dataset describtion page of github"),

tags$a(href="https://www.kaggle.com/maitree/wine-quality-selection", "Click Here!"),

hr(),

# Sidebar layout

sidebarLayout(

# Sidebar panel

sidebarPanel(

# Choosing dataset

radioButtons(inputId = "dataset",

label = "Choose a wine type",

choices = list("Red Wine", "White Wine"), selected = NULL, inline = FALSE),

# The number of obs to view

numericInput(inputId = "obs",

label = "Number of observations to view:",

value = 5)

),

# Main panel

mainPanel(

tabsetPanel(

tabPanel("Summary&Data\_View",

tags$h5("In this tab, you can see the summary and rows of the dataset you chose"),

hr(),

# Output summary

verbatimTextOutput("summary"),

# Output data set table

tableOutput("view")),

tabPanel("Quality Plot",

tags$h5("In this tab, you can see the plots for wine quality"),

hr(),

# Output plot1

plotOutput("quality\_plot"))

)

)

)

)

# Server

server <- function(input, output) {

# Dataset input

datasetInput <- reactive({

switch(input$dataset,

"Red Wine" = red,

"White Wine" = white)

})

# Output

#Output : summary of overall

output$summary <- renderPrint({

dataset <- datasetInput()

summary(dataset)

})

#Output : view of dataset with selected condition

output$view <- renderTable({

data <- datasetInput()

head(data, n = input$obs)

})

library(ggplot2)

library(dplyr)

output$quality\_plot <- renderPlot({

data <- datasetInput()

p1 <- ggplot(data=data, aes(x=quality, y=pH)) + geom\_point()

p1

})

}

shinyApp(ui = ui, server = server)

```

## Feature selection

```{r reg, echo=FALSE}

#feature selection preparation

wine\_reg\_int <- lm(quality ~1, data=wine)

wine\_reg\_all <- lm(quality ~ ., data=wine)

summary(wine\_reg\_all)

```

## Stepwise

```{r step\_f, echo=FALSE}

library(MASS)

wine\_reg\_all\_f <- formula(wine\_reg\_all)

wine\_reg\_int\_f <- formula(wine\_reg\_int)

#forward

wine\_forward <- stepAIC(wine\_reg\_int, scope=wine\_reg\_all\_f, direction="forward")

wine\_forward$anova

```

```{r step\_b, echo=FALSE}

#backward

wine\_back <- stepAIC(wine\_reg\_all, scope=wine\_reg\_int\_f, direction="backward", trace=F)

wine\_back$anova

```

##Regression

```{r linear1, echo=FALSE}

wine\_reg\_fin <- lm(quality ~ fixed.acidity + volatile.acidity + residual.sugar +

chlorides + free.sulfur.dioxide + total.sulfur.dioxide +

density + pH + sulphates + alcohol, data=wine)

summary(wine\_reg\_fin)

```

##Cross Validation

Compare the full model (the linear regression fit that contains all of the variables) and your model in step using cross validation (use K-fold with K=10).

```{r cv, echo=FALSE}

library("boot")

full\_glm <- glm(wine\_reg\_all)

cv.glm(data=wine, glmfit = full\_glm, K=10)$delta[1]

fit\_glm <- glm(wine\_reg\_fin)

cv.glm(data=wine, glmfit = fit\_glm, K=10)$delta[1]

```

##LASSO

```{r lasso, echo=FALSE}

library(glmnet)

lasso.cv <- cv.glmnet(x=as.matrix(wine[,-12]),y=as.matrix(wine[,12]),alpha=1,nfolds = 10)

plot(lasso.cv)

lasso.cv$lambda.min

min(lasso.cv$cvm)

```

```{r lasso, echo=FALSE}

library("car")

residualPlots(wine\_reg\_fin)

```

##Adding Nonlinearity

#interaction terms

```{r inter, echo=FALSE}

inter.wine<-lm(quality ~ fixed.acidity\*volatile.acidity\*pH + residual.sugar +

chlorides + free.sulfur.dioxide + total.sulfur.dioxide +

density + sulphates + alcohol, data=wine)

summary(inter.wine)

inter.wine\_glm <- glm(inter.wine)

cv.glm(data=wine, glmfit = inter.wine\_glm, K=10)$delta[1]

```

```{r inter1, echo=FALSE}

inter.wine2<-lm(quality ~ fixed.acidity\*pH+volatile.acidity + residual.sugar +

free.sulfur.dioxide + total.sulfur.dioxide +

density + sulphates + alcohol, data=wine)

summary(inter.wine)

inter.wine\_glm2 <- glm(inter.wine2)

cv.glm(data=wine, glmfit = inter.wine\_glm2, K=10)$delta[1]

```

```{r inter2, echo=FALSE}

inter.wine3<-lm(quality ~ fixed.acidity\*pH+volatile.acidity + residual.sugar +

free.sulfur.dioxide\*total.sulfur.dioxide +

density + sulphates + alcohol, data=wine)

summary(inter.wine3)

inter.wine\_glm3 <- glm(inter.wine3)

cv.glm(data=wine, glmfit = inter.wine\_glm3, K=10)$delta[1]

```

```{r inter3, echo=FALSE}

inter.wine4<-lm(quality ~ fixed.acidity\*pH+volatile.acidity + residual.sugar +

free.sulfur.dioxide\*total.sulfur.dioxide\*sulphates+

density + alcohol, data=wine)

summary(inter.wine4)

inter.wine\_glm4 <- glm(inter.wine4)

cv.glm(data=wine, glmfit = inter.wine\_glm4, K=10)$delta[1]

```

#splines

```{r gam, echo=FALSE}

install.packages("gamclass")

library("gamclass")

library("mgcv")

gam.wine <- gam(quality ~ fixed.acidity + volatile.acidity + residual.sugar +

chlorides + s(free.sulfur.dioxide) + total.sulfur.dioxide +

density + pH + sulphates + alcohol, data=wine)

summary(gam.wine)

CVgam(formula=quality ~ fixed.acidity + volatile.acidity + residual.sugar +

chlorides + s(free.sulfur.dioxide) + total.sulfur.dioxide +

density + pH + sulphates + alcohol, data = wine, nfold = 10)$cvscale

```

```{r gam\_inter, echo=FALSE}

inter.wine\_gam<-gam(quality ~ fixed.acidity\*pH+volatile.acidity + residual.sugar +

free.sulfur.dioxide\*total.sulfur.dioxide\*sulphates+

density + alcohol, data=wine)

summary(inter.wine\_gam)

CVgam(formula=quality ~ fixed.acidity\*pH+volatile.acidity + residual.sugar +

free.sulfur.dioxide\*total.sulfur.dioxide\*sulphates+

density + alcohol, data = wine, nfold = 10)$cvscale

```

#transforming

```{r transform, echo=FALSE}

rmMSE <- rep(0,10)

for(i in 1:10){

templm <- glm(quality ~ fixed.acidity\*volatile.acidity\*pH + residual.sugar +

chlorides + poly(free.sulfur.dioxide,i) + total.sulfur.dioxide +

density + sulphates + alcohol, data = wine)

tempCV <- cv.glm(wine,templm,K = 10)

rmMSE[i] <- tempCV$delta[1]

}

plot(rmMSE)

which.min(rmMSE)

```

```{r transform\_glm}

trans\_GLM <- glm(quality ~ fixed.acidity\*pH+volatile.acidity + residual.sugar +

poly(free.sulfur.dioxide,3)\*total.sulfur.dioxide\*sulphates+

density + alcohol,data=wine)

summary(trans\_GLM)

cv.glm(wine,trans\_GLM,K=10)$delta[1]

```

##Bootstrap

Compare confidence intervals from normal approximation to bootstrap confidence intervals.

```{r boot, echo=FALSE}

#### BOOTSTRAP ####

N <- 6497

x <- rnorm(N)

lm1 <- inter.wine4

confint(lm1)

lmResid <- residuals(inter.wine4)

tempBeta <- c()

for(i in 1:10000){

lmResidResampled <- sample(lmResid,replace = TRUE)

tempY <- coef(lm1)[1] + x\*(coef(lm1)[2]) + lmResidResampled

tempLm <- lm(tempY~x)

tempBeta[i] <- coef(tempLm)[2]

}

hist(tempBeta-(coef(lm1)[2]),50)

c((coef(lm1)[2])-1.96\*sd(tempBeta),(coef(lm1)[2])+1.96\*sd(tempBeta))

quantile(tempBeta,c(.025,.975))

confint(lm1)

library(boot)

# function to obtain regression weights

bsa <- function(formula, data, indices) {

d <- data[indices,] # allows boot to select sample

fit <- lm(formula, data=d)

return(coef(fit))

}

bsa(inter.wine4, wine)

results <- boot(data=wine, statistic=bsa, R=10000, formula=inter.wine4)

results

boot.ci(results, type=c("bca", "norm"))

```

## Evaluation

```{r evaluation1, echo=FALSE}

plot(inter.wine4)

```

##Huge package

```{r huge, echo=FALSE}

install.packages("huge")

library("huge")

wine.m <- as.matrix(wine)

wine\_huge <- huge(wine.m, lambda = NULL, nlambda = NULL, lambda.min.ratio = NULL, method = "mb", scr = NULL, scr.num = NULL, cov.output = FALSE, sym = "or", verbose = TRUE)

plot.huge(wine\_huge)

```

```{r split}

##Sample the dataset. The return for this is row nos.

row.number <- sample(1:nrow(wine), 0.7\*nrow(wine))

train = wine[row.number,]

test = wine[-row.number,]

dim(train)

dim(test)

```

```{r train}

tain\_model <- lm(quality ~ fixed.acidity\*pH+volatile.acidity + residual.sugar +

free.sulfur.dioxide\*total.sulfur.dioxide\*sulphates+

density + alcohol, data=train)

summary(tain\_model)

tain\_model\_glm <- glm(tain\_model)

cv.glm(data=train, glmfit = tain\_model\_glm, K=10)$delta[1]

```

```{r test}

pred1 <- predict(inter.wine4, newdata = test)

rmse <- sqrt(sum((exp(pred1) - test$quality)^2)/length(test$quality))

c(RMSE = rmse, R2=summary(inter.wine4)$r.squared)

```