Case Study 1

Linear Methods Application to a Real World Data Set (Regression)

Submitted by

Md Salman Rahman

Graduate Student

University of Texas Rio Grande Valley

Submitted to

Dr. Tamer Oraby

October 20, 2021

Contents

1 Data Preparation

1.1 Descriptive Analysis

This data set is about Tinnitus which is one of the most distressing hearing related symptoms. The data set contains 142 subjects consisting clinical and demographic factors along with pre and post TFI (14 columns). Figure 1 represent the summary statistics of the Tinnitus data where we can see Subject_ID is about description. Also, we can see that the mean Pre_TFI_Score and Post_TFI_Score are 59.37 and 35.41 respectively.

```
> summary(tinni_data_frame)
  Subject_ID           Group            HHI_Score          GAD              PHQ              ISI
 Length:142         Length:142         Min.   : 0.00   Min.   : 0.000   Min.   : 0.000   Min.   : 0.00
 Class :character   Class :character   1st Qu.: 8.00   1st Qu.: 3.000   1st Qu.: 4.000   1st Qu.: 8.00
 Mode  :character   Mode  :character   Median :18.00   Median : 6.000   Median : 7.000   Median :13.00
                                       Mean   :17.79   Mean   : 7.479   Mean   : 8.028   Mean   :12.96
                                       3rd Qu.:26.00   3rd Qu.:11.000   3rd Qu.:11.000   3rd Qu.:18.00
                                       Max.   :40.00   Max.   :21.000   Max.   :27.000   Max.   :27.00

      SWLS        Hyperacusis         CFQ             Gender           Age        Duration_of_tinnitus(years)
 Min.   : 5.00   Min.   : 1.00   Min.   : 7.00   Min.   :1.000   Min.   :22.00   Min.   : 0.30
 1st Qu.:14.00   1st Qu.:13.00   1st Qu.:29.25   1st Qu.:1.000   1st Qu.:46.25   1st Qu.: 3.00
 Median :20.00   Median :18.50   Median :41.00   Median :1.000   Median :58.00   Median :10.00
 Mean   :20.32   Mean   :19.04   Mean   :40.59   Mean   :1.437   Mean   :55.45   Mean   :11.99
 3rd Qu.:26.00   3rd Qu.:25.00   3rd Qu.:50.00   3rd Qu.:2.000   3rd Qu.:65.00   3rd Qu.:15.00
 Max.   :35.00   Max.   :42.00   Max.   :86.00   Max.   :2.000   Max.   :83.00   Max.   :55.00

 Pre_TFI_Score   Post_TFI_Score
 Min.   :24.40   Min.   : 4.00
 1st Qu.:46.80   1st Qu.:18.50
 Median :58.60   Median :29.60
 Mean   :59.37   Mean   :35.41
 3rd Qu.:73.60   3rd Qu.:52.80
 Max.   :97.20   Max.   :88.40
                 NA's   :86
```

Figure 1: Summary statistics of Tinnitus data

Histogram of all the continuous and categorical variable is drawn (available in R code), to get some preliminary description about the data. From the histogram analysis we find that GAD, PHQ, Duration_of_tinnitus(years), Post_TFI_Score are right skewed and Age is left skewed. On the other side, ISI, SWLS, Hyperacusis, CFQ, Pre_TFI_Score are symmetric. Also, pie chart for gender show that there are 80 male and 62 female person in the data set.

Correlation matrix, scatterplots for several pairs of variables are drawn (available in R code). From the correlation analysis we find that correlation coffecient of GAD and PHQ is 0.76, wheras for Pre_TFI_Score and Post_TFI_Score the value is 0.60,

and the correlation coefficient of PHQ and Pre_TFI_Score is 0.64. Details correlation analaysis is available in the R code.

1.2 Handling Missing Value

There are 86 missing (NA) value (for Post_TFI_Score) in the data set (Figure 1). Data without missing value can be summarized by some descriptive statistical measure such as mean, mode, or variance. One of the simplest way to to impute missing values is to fill in such a way that descriptive statistical remain same. So, we fill the missing value of Post_TFI_Score with mean of Post_TFI_Score (mean=35.40714). After the data imputation, a new variable called TFI_Reduction is created by subtracting Pre_TFI_Score from Post_TFI_Score. TFI_Reduction will be our response/target in multiple regression modeling. Subject_ ID, Pre_TFI_Score, and Post_TFI_Score removed from the data frame as these are not required for further step.

1.3 Multicollinearity

We see from Figure 2 some co-relation between variable. For example, GAD and PHQ, PHQ and ISI, CFQ and Hypercacusis are showing some collinearlity. And the Figure 2 gives much broader visualization.

1.4 Training and Testing Data

We split our final data set into 80% training and 20% testing set by using sample() function of base package and set.seed(123) is considered to make the partition reproducible.

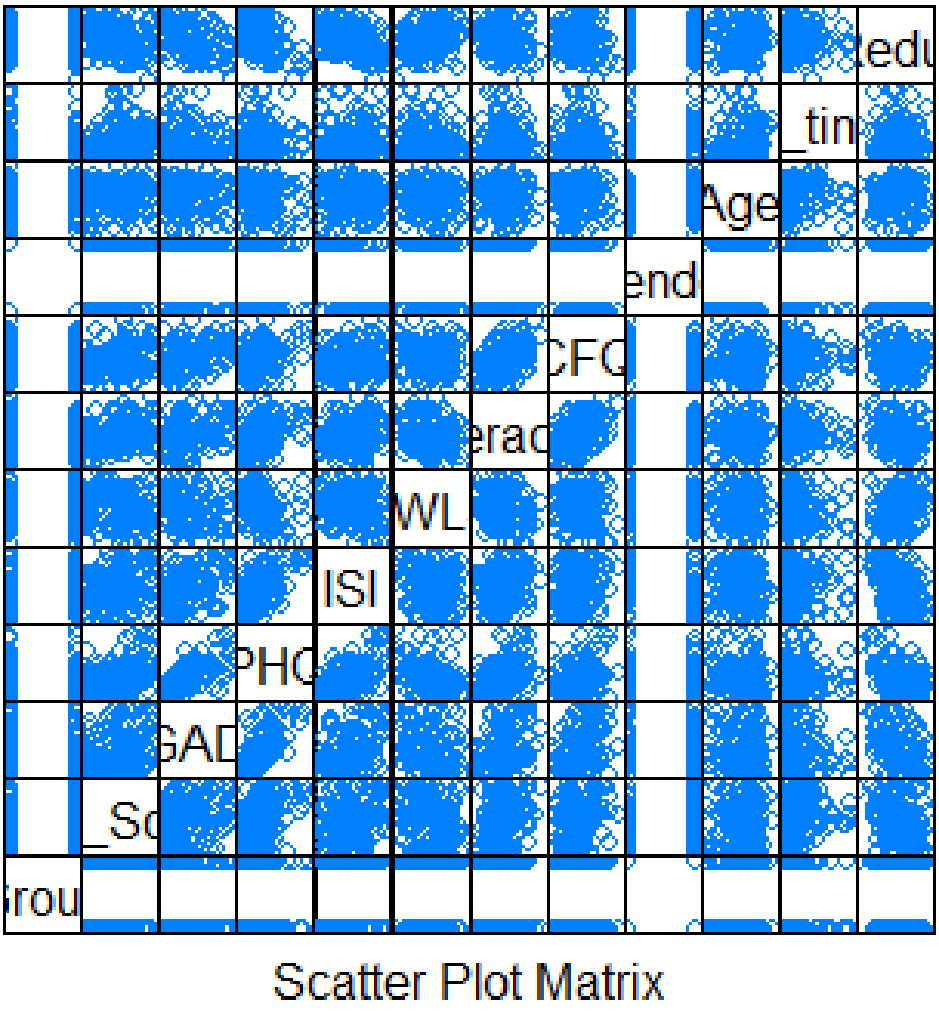

Figure 2: Collinearity Check

2 Regression Analysis

2.1 Multiple Linear Regression

This is a multiple linear regression problem as we have multiple regressor to predict the TFI_Reduction. The linear general model will be:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon \tag{1}$$

where, Y = response, $X_1, X_2 \cdots X_p$ = predictors, $\beta_1, \beta_2, \cdots, \beta_p$ = Coefficient/parameter corresponding to regressor, and ϵ = error term.

Initially, I run the full model containing all the regressor and check the coefficient, their corresponding p value, and also identify potential outlier by checking the scale location, residuals vs leverage, and normal Q-Q plot. After identifying the outlier (data-point: 30,32,58), I remove them. Then, I run the full model again with new data after removing the outliers. Adjusted R^2 improve after removing the outliers, this is not the final conclusion. I tried different model by removing the regressor with high p value. For testing purpose, I develop separate model for example one with (Group + HHI_Score + Gender + ISI + SWLS) and test them with anova and find that the model is not significant at 95% significance level. I also check interaction within the regressor and they didn't show any significant result.

Note: Outlier is removed for experimental purpose, and as it is a statistical machine learning work, I didn't want to remove outlier blindly because removing most of the outlier leads the model to over fit and create some problem during testing.

2.1.1 Subset Selection

Subset selection is a technique where we identify a subset of predictors which is related to the response Y. After selecting the subset, we then fit the model with

least square which gives us some advantage as the variable is reduced during the subset selection. There are two approach for selecting the subset, (1) Best Subset Selection and (2) Stepwise (Forward and Backward) Model Selection.

1. **Best Subset Selection**: Let us consider a null model which contain no predictors and we will simply predict the sample mean for each observation. After that we will fit exactly $\binom{p}{k}$ (say we have p predictors) and $k = 1, 2, 3, \cdots p$ and we will pick the best model which have the smallest RSS or largest R^2. Finally, we will choose final single best model according to the cross-validated prediction error, C_p, AIC, BIC, or adjusted R^2.

2. **Stepwise Selection**: One of the disadvantage of best subset selection is, it cannot be applied when p value is very large (for computational reason and statistical problems with large p value).

    - **Forward Stepwise Selection**: Forward stepwise selection start with model with no predictors and it add predictor in one at a time until all predictors are in the model. At each step the model which provide greatest additional improvement will consider as final selected model.
    - **Backward Stepwise Selection**: Unlike the forward stepwise selection, the backward stepwise selection start with full least square model with all the predictors, and it remove the least useful predictor one at a time.

For our model, I use backward, exhaustive, sequentially add and replace (seqrep), and forward subset selection using the library leaps (reg subsets function) and figure 5 shows the various combination of subsets for forward selection.
I use step function of R to generate the result and step function use AIC for the selection. Also, we can able to use step function using BIC by changing the k value (k=log (number of data point) for BIC, and k=2 for AIC). Also, in the step function we can able to change whether the direction will be backward, forward or both. I

```
Call:
lm(formula = TFI_Reduction ~ HHI_Score + ISI + SWLS, data = training)

Residuals:
    Min      1Q  Median      3Q     Max
-33.820 -10.291  -0.831   9.722  44.083

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -19.4014     6.1011  -3.180  0.00192 **
HHI_Score    -0.2904     0.1339  -2.169  0.03229 *
ISI          -0.7424     0.2210  -3.360  0.00108 **
SWLS          0.5140     0.2101   2.446  0.01602 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.9 on 109 degrees of freedom
Multiple R-squared:  0.225,     Adjusted R-squared:  0.2037
F-statistic: 10.55 on 3 and 109 DF,  p-value: 3.79e-06
```

Figure 3: Stepwise Selection Result

```
> l_model_step_AIC$anova
                                 Step Df   Deviance Resid. Df Resid. Dev      AIC
1                                     NA         NA       101   26331.99 639.9802
2                           - Group  1   1.122663       102   26333.11 637.9850
3                             - PHQ  1  41.212199       103   26374.32 636.1617
4 - `Duration_of_tinnitus(years)`  1  54.374784       104   26428.70 634.3944
5                      - Hyperacusis  1  94.962801       105   26523.66 632.7997
6                             - Age  1 159.680938       106   26683.34 631.4780
7                          - Gender  1 258.900176       107   26942.24 630.5691
8                             - CFQ  1 202.573849       108   27144.82 629.4156
9                             - GAD  1 420.776430       109   27565.59 629.1538
> |
```

Figure 4: Anova of Stepwise Selection

inspect all the possible way and the result of best one is inserted in figure 3 and 4. I also explore the anova of the result and figure 4 explain the anova result where we can able to see the AIC value of the variable which was removed in the subset selection.

```
  Group1 HHI_Score GAD PHQ ISI SWLS Hyperacusis CFQ Gender2 Age `Duration_of_tinnitus(years)`
1  ( 1 )                  *
2  ( 1 )                  *   *
3  ( 1 )                  *   *    *
4  ( 1 )              *   *   *    *
5  ( 1 )          *   *   *   *    *
6  ( 1 )          *   *   *   *    *                          *
7  ( 1 )          *   *   *   *    *              *           *
8  ( 1 )      *   *   *   *   *    *              *           *
```

Figure 5: Subset

Figure 3 and 4 suggest TFI_Reduction $\sim$ HHI_Score + ISI + SWLS as best subset among the different subset.

We generate crossvalidation MSE using lars library and cv.lars function and the plot shown in Figure 6. From the figure we see that we have minimum cross

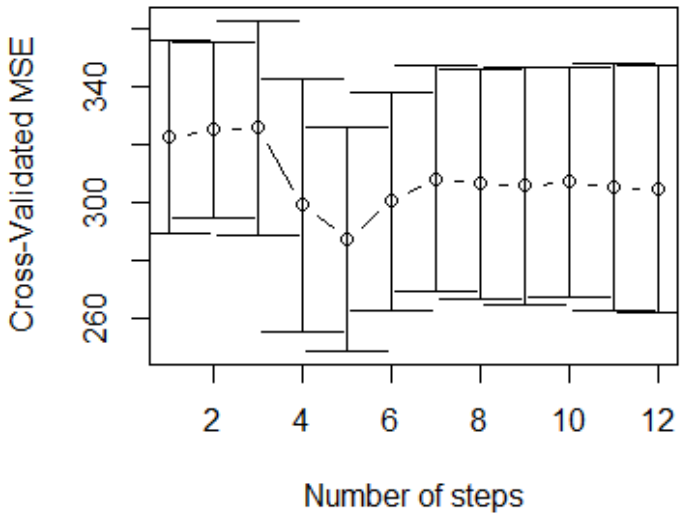

Figure 6: Cross Validation MSE Vs. Number of Steps

validation error between 4 and 6 and its gives a strong hints about selecting model with high predictive power.

Figure 5 shows the summary of the best model inside each subset size. It means if I want to include only one variable then I will choose the PHQ, and if I want to include two variable then it will be PHQ and ISI, and so on. This figure 5 also helps me to get more insights about the model selection.

2.1.2 Model Diagnostics

There are several approach for the diagnosis of the model. For this case study, I use adjusted R^2, Mallows's Cp, RSS, and BIC. Adjusted R^2 of 8 possible combination of subset (Figure 5) are 0.1284271, 0.1593308, 0.1875721, 0.2008046, 0.2012170, 0.2010023, 0.1987952, 0.1980581 respectively. And we see that number 5 have the largest adjusted R^2.

Also the residual sum of squares are 30725.03, 29368.61, 28123.99, 27412.10,

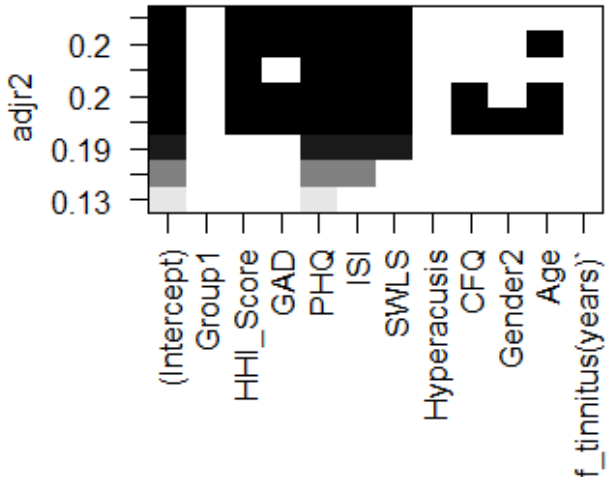

Figure 7: Plot of Adjusted R^2

27144.27, 26897.81, 26717.66, 26487.55 for all the 8 model (Figure 5). And, it is obvious that the number 8 will give the smallest RSS because adding more variable in the model will reduce the RSS. But, this leads to overfiting.
Mallows's Cp for the 8 model are 8.850114, 5.647384, 2.873463, 2.142922, 3.115625, 4.170295, 5.479300, 6.596684. And the number 4 give the smallest mallow's Cp.

Also, I inspect BIC and this are value for 8 model -7.091183, -7.465875, -7.631788, -5.801530, -2.183635, 1.513069, 5.481081, 9.231027. Model 3 in the Figure 5 gives the smallest BIC value.
I make a result combining the result of adjusted R^2, RSS, BIC, and Cp showed in figure 10. We can see that all the the four evaluation method approved this three regressor PHQ, ISI, and SWLS. We can able to connect all the previous result and

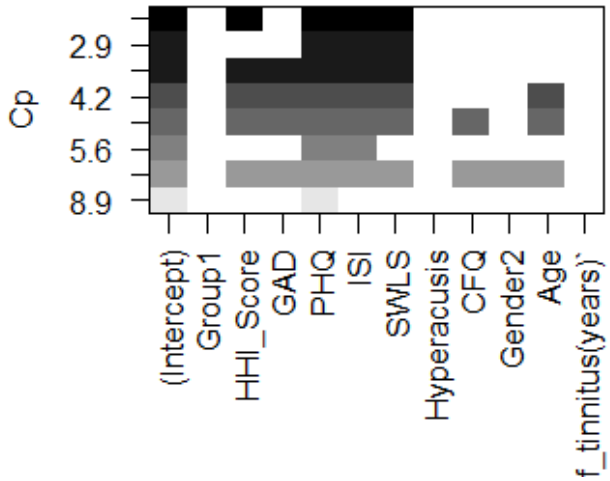

Figure 8: Plot of Cp

make some decision about the final model. This is strongly connected to previous result as we saw something very close in the forward stepwise selection and best subset selection. The assumption of linear regression such as normality of residuals, homogeneity of residual variance, independence of residuals error terms also checked by plotting the residual (using plot() or autoplot() function in R) and fitted value. And, there is no major violation of the assumption. And the model TFI_Score ~HHI_Score + ISI + PHQ + SWLS seems good.

2.1.3 Final Model

Considering all the inspection, I fit various model using the best selection obtained from adjusted R^2, BIC, mallows Cp and initally I decided combination of HHI_Score + ISI + PHQ + SWLS as our best model. But I check the p value and it is high for PHQ. So, I fit another model with HHI_Score + ISI + SWLS. It is perfect from all

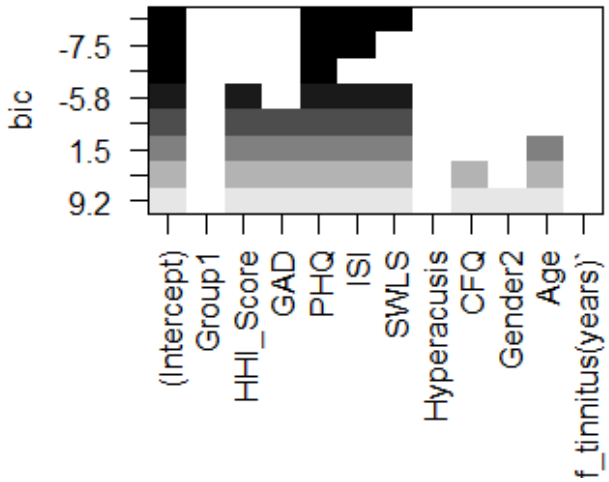

Figure 9: Plot of BIC

	adjr2	rss	cp	bic
(Intercept)	TRUE	TRUE	TRUE	TRUE
Group	FALSE	FALSE	FALSE	FALSE
HHI_Score	TRUE	TRUE	TRUE	FALSE
GAD	TRUE	TRUE	FALSE	FALSE
PHQ	TRUE	TRUE	TRUE	TRUE
ISI	TRUE	TRUE	TRUE	TRUE
SWLS	TRUE	TRUE	TRUE	TRUE
Hyperacusis	FALSE	FALSE	FALSE	FALSE
CFQ	FALSE	TRUE	FALSE	FALSE
Gender	FALSE	TRUE	FALSE	FALSE
Age	FALSE	TRUE	FALSE	FALSE
Duration_of_tinnitus(years)	FALSE	FALSE	FALSE	FALSE

Figure 10: Combine Result of adjusted R^2, RSS, BIC, Cp

point of view and earlier evidence also support this. This model is not very complex and interpret able and give the best output.

$$TFI\_Reduction = \beta_0 + \beta_1 * HHI\_Score + \beta_2 * ISI + \beta_3 * SWLS \tag{2}$$

The value of the coefficient $\beta_0 = -19.4014, \beta_1 = -0.2904, \beta_2 = -0.7424, \text{and} \beta_3 = 0.5140$.

2.1.4 Prediction

In earlier step we separate the testing data and test our final model with testing data. I perform the prediction and the mean square error on the testing is 261.8514.

2.2 Ridge Regression

Ridge and Lasso are the shrinkage methods where the model contain all the p predictors but the approach shrinks the coefficient estimates towards zero. Shrinking the coefficient significantly reduce the variance. The ridge regression coefficient estimates $\hat{\beta}^R$ which minimize

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} (\beta_j x_{ij}) \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{3}$$

In our case study, I use glmnet library and glmnet function to perform the ridge regression. In the function we use alpha = 0 for ridge regression. In the glmnet package, the standardization is define as true and we use the lambda value as a list of 100 number from 100 to 0.01. I also perform prediction for some particular λ for experimental purpose and getting some insights about ridge. In figure 11, we see the mean and standard deviation for λ. Here, we notice two vertical line, one is lambda.min and another is lambda.1se and the value are 22.46 and 702.115. The first value is the ultimate minimum for λ and another is one standard error from the ultimate minimum. If we take the logarithm of this two, we will get the position of

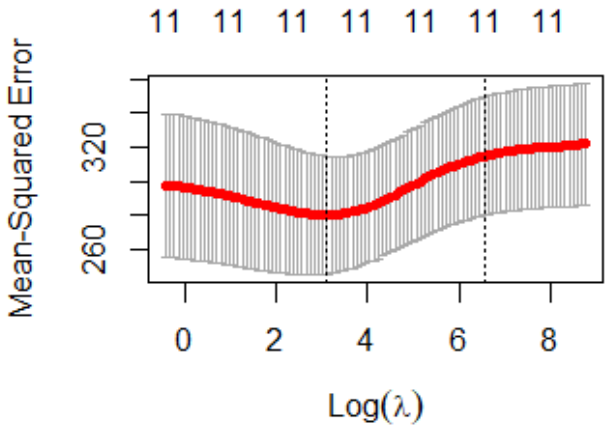

Figure 11: Optimal λ Value for Ridge

vertical line in the graph. I also check the effective degree of freedom using this two different lambda value. Considering, all the fact, I choose ultimate minimum value 22.46 as the effective lambda value and it also give the smallest error than the other one. Finally, I make prediction and the mean square error is 310.9943.

2.3 Lasso Regression

One disadvantage of ridge regression include all p predictors in the model which motivates the discovery of lasso. The lasso coefficient $\hat{\beta}_\lambda^L$

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}(\beta_j x_{ij})\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \tag{4}$$

I perform the lasso in our case study data and the process is exactly same which I describe in section 2.1.5 Ridge Regression but the only difference is that here we

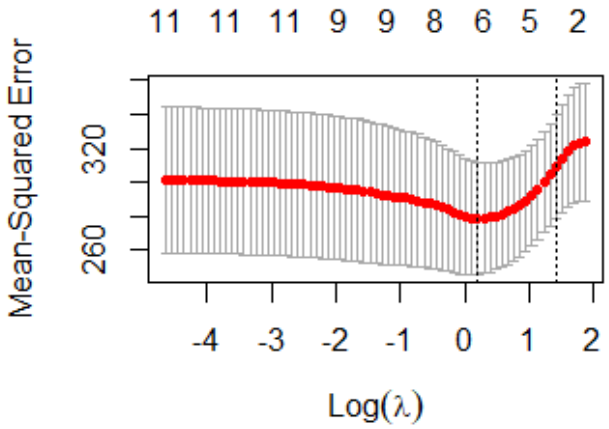

Figure 12: Optimal λ Value for Lasso

use alpha = 1 for lasso. Figure 12 shows the optimal λ value for lasso. The optimal lambda value for lasso is 1.226967 (ultimate), and the mean square error is 280.2552.

2.4 Principal Component Regression (PCR)

Principal component regression is a dimensionality reduction method which transform the predictors to fit a least squares model using the transformed variables. For example, first principal component is the linear combination of the variable that contain the variable which have the largest variance. In figure 13, we see that % variance explained using PCR. For example, if we include one principal component, it will explain 29.21 % of the total variability in the input. Similarly if we include 6 component it will explain 78.69 % total variability of the input and so on. Also, figure 14, we saw that we can choose optimal number of component between 5-6. Also, I make the same plot for training and there is a elbow at the point 4, so I use

```
> summary(pcr.model)
Data:   X dimension: 142 11
        Y dimension: 142 1
Fit method: svdpc
Number of components considered: 11

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps
CV           18.97    16.61    16.69    16.72    16.68    16.35    16.32    16.45    16.59    16.71     16.76
adjCV        18.97    16.60    16.67    16.71    16.71    16.30    16.28    16.42    16.54    16.65     16.70
       11 comps
CV        16.85
adjCV     16.79

TRAINING: % variance explained
               1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps
X                29.21    42.50    52.92    62.30    71.18    78.69    85.12    90.18    94.63     98.33
TFI_Reduction    24.13    24.15    24.34    26.11    29.45    30.12    30.12    30.71    30.72     30.76
               11 comps
X                100.00
TFI_Reduction     31.01
```

Figure 13: Summary of PCR Result

number of components as 4. And, this model explain around 62.30% variability of the total input. Along with this, I check the projections, coefficient, and loading component of the PCR model. The PCR model is trained and the mean square error for PCR is 278.405.

Note: pcr package of R from pls library is used for the analysis of result.

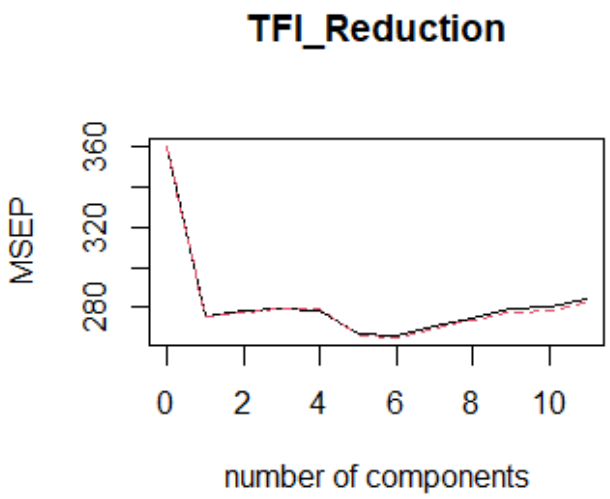

Figure 14: PCR Optimal Number of Component Plot

2.5 Partial Least Square Regression (PLSR)

One of the problem with principal component regression is that it identify the linear combination of variable in unsupervised way as the response is not used to identify the principal component. Partial least square regression solve the problem by doing this process in supervised way. Partial least square first choose a new set of feature which is linear combination of original feature and then fit a linear model via ordinary least square using the new features.

For our case study, I use plsr function is used to make partial least square regression.

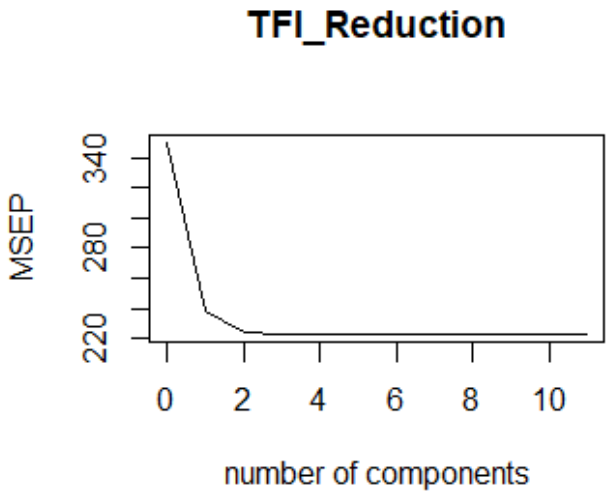

Figure 15: PLSR Optimal Number of Component Plot

From figure 15, we see that the optimal number of component vary from 2 to 3. I choose 2 component, and build the model. Finally, I make the prediction with PLSR model and the mean square error for partial least square regression is 285.6339.

3 K-means Regression

K-means regression is a non-parametric method. Non-parametric methods are very flexible since they do not follow any certain form but it suffer from overfitting and required big data as large number of parameter involved. In our case study, we use knn.reg function from the library class to build the K-means regression model.

3.1 Monte-Carlo Cross Validation for Selecting Best k Value

Figure 16 depicts the k value vs cross validation and we find that optimal K value is 22 for the case study. After that, the model is trained and tested considering K=22.

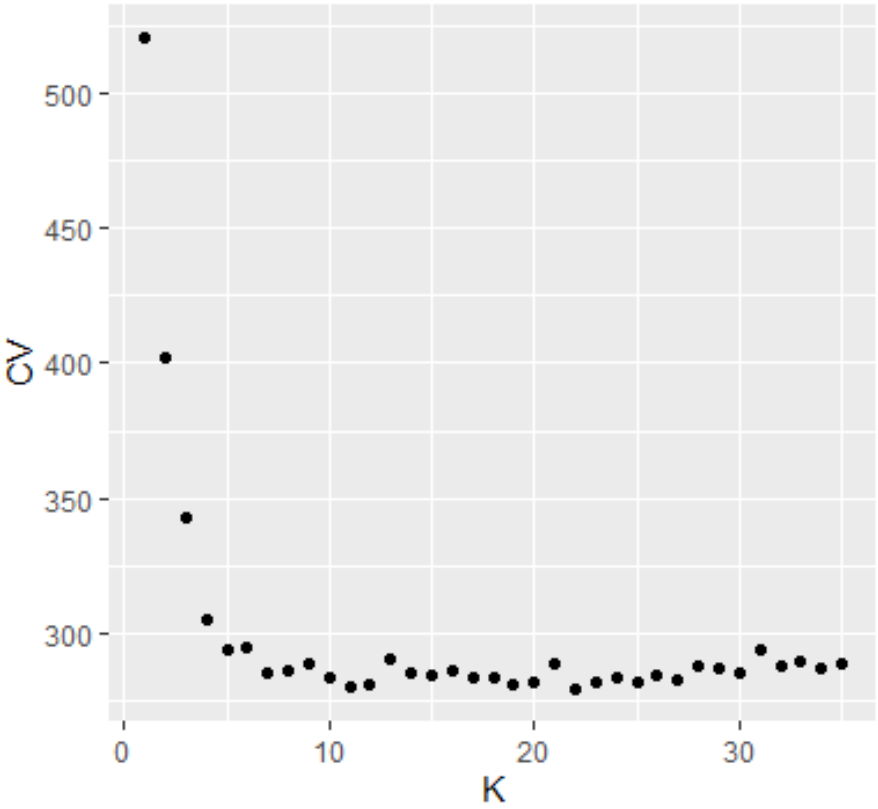

Figure 16: Optimal K value

3.2 Prediction

For measuring how far the predict value from the actual value, I use squared error loss function as our response is continuous and the mean square error is 442.0846.

4 Comparing Multiple and K-means Regression

Among all the model the multiple regression with TFI_Reduction $\sim$ HHI_Score + ISI + SWLS gives the smallest mean square error value. And, the model is simple and interpretable. Between lasso and ridge, the lasso model perform better and it make sense also. Because, we know that lasso is more about sparsity and ridge is more applicable for dense model.

5 R Code

```
"
Author: Md Salman Rahman
Course: MATH 6333 Statistical Learning
Course Instructor: Dr. Tamer Oraby
Case Study 1
"

# Solution

###################### Multiple Regression Analysis ################################

## reading the data set
```

```
library("readxl")
library(ggplot2)
tinnitus_data<- read_excel("C:/Users/User/OneDrive -
The University of Texas-Rio Grande Valley/Course_video/
Statistical Learning/MATH 6333 90L - SL - F21/Homework_salman/Case Study 1/CaseStud
tinni_data_frame<-as.data.frame(tinnitus_data)




### Step 1:
# exploring the data set by performing some descriptive analysis

#get means excluding the missing value
sapply(tinni_data_frame, mean, na.rm=TRUE)

#summary
summary(tinni_data_frame)




# histogram
hist(tinni_data_frame$HHI_Score)
hist(tinni_data_frame$GAD)
hist(tinni_data_frame$PHQ)
hist(tinni_data_frame$ISI)
hist(tinni_data_frame$SWLS)
hist(tinni_data_frame$Hyperacusis)
```

```
hist(tinni_data_frame$CFQ)
hist(tinni_data_frame$Gender)
hist(tinni_data_frame$Age)
hist(tinni_data_frame$`Duration_of_tinnitus(years)`)
hist(tinni_data_frame$Pre_TFI_Score)
hist(tinni_data_frame$Post_TFI_Score)


# pie chart
mytable<-table(tinni_data_frame$Gender) # 1 is Male and 2 is Female
pie(mytable)

# correlation analysis
library(corrplot)


# correlation matrix
corrplot(cor(tinni_data_frame[,3:14]),
         method = "number",
         type = "upper" # show only upper side
)

# scatterplot of several variable
pairs(tinni_data_frame[,3:14])

# correlation tests for whole data set
library(Hmisc)
res <- rcorr(as.matrix(tinni_data_frame[,3:14])) # rcorr() accepts matrices only
```

```
# correlation test
library(correlation)

correlation::correlation(tinni_data_frame[,3:14],
                         include_factors = TRUE, method = "auto"
)


### Step 2 and 3:
#checking the missing value in the Pre and Post TFI scores

table(is.na(tinni_data_frame))


# if there is missing value use data imputation with
mean corresponding to the Pre or Post TFI score


# If there are any numerical measurement with missing values,
we will use mean to impute the data
# and if there are any missing values for any categorical measurement,
we will use mode to impute the data.


tinni_data_frame$Post_TFI_Score[is.na(tinni_data_frame$Post_TFI_Score)]
<- mean(tinni_data_frame$Post_TFI_Score, na.rm = TRUE)
```

```
# after the data imputation please create a new
variable called TFI_Reduction by subtracting
# "Post_ TFI_Score" by "Pre_TFI_Score".
# we will use the TFI_Reduction as the response
in our multiple linear regression model

tinni_data_frame$TFI_Reduction <-
(tinni_data_frame$Post_TFI_Score - tinni_data_frame$Pre_TFI_Score)



#removing pre and post TFI score and subject ID

tinni_data_frame = subset(tinni_data_frame, select =
-c(Subject_ID, Pre_TFI_Score,Post_TFI_Score) )


##baseline



# replacing male and female with 1 and 0 respectively

tinni_data_frame$Group[tinni_data_frame$Group == "Treatment"] <- 1
```

```
tinni_data_frame$Group[tinni_data_frame$Group == "Control"] <- 0

tinni_data_frame$Gender[tinni_data_frame$Gender == 1] <- 1
tinni_data_frame$Gender[tinni_data_frame$Gender == 2] <- 0

tinni_data_frame$Group <- as.numeric(tinni_data_frame$Group)

View(tinni_data_frame)

# converting the group and Gender into factor as they are categorical
#tinni_data_frame$Gender <- as.factor(tinni_data_frame$Gender)
#contrasts(tinni_data_frame$Gender)

#Multicollinearity check

library(lattice)
splom(tinni_data_frame[,],pscales=0)
round(cor(tinni_data_frame[c(-1,-9)]),2)



## Step 4:
# partitioning the data set (obtained at step 2)
into training (80% of the data ) and a test set
# Use set.seed(123) and sample() in base package or
createDatapartition() in caret package

set.seed(123) # to make your partition reproducible
```

```
sample_size <- floor(0.80 * nrow(tinni_data_frame))

train_ind <- sample(seq_len(nrow(tinni_data_frame)), size = sample_size)

training <- tinni_data_frame[train_ind, ]
testing <- tinni_data_frame[-train_ind, ]

training <- as.data.frame(training)



### Step 5:
# Performing the multiple regression analysis
on the data using the TFI_Reduction of each subject
# as the response. There will be some comment on finding.
# Tips: we can use best subset selection/forward/backward
selection methods to select the best multiple
# linear regression model with lm() in R


full_ls <- lm(TFI_Reduction ~ ., data=tinni_data_frame)
summary(full_ls)
plot(full_ls)

#removing outlier

tinni_d <-tinni_data_frame[-c(3,32,57,58),]
full_l <- lm(TFI_Reduction ~ ., data=tinni_data_frame)
```

```
summary(full_1)

# new model
full_1_2 <- lm(TFI_Reduction ~ Group +
HHI_Score + ISI + SWLS, data=tinni_data_frame)
summary(full_1_2)

# new model
full_1_3 <- lm(TFI_Reduction ~ HHI_Score + ISI + SWLS +Gender , data=tinni_d)
summary(full_1_3)


# anova test of this two model
anova(full_1,full_1_2)
fitted(full_1_2)
resid(full_1_2)

# anova for test 3
anova(full_1,full_1_3)


# model with some interaction
full_1_4 <- lm(TFI_Reduction ~ HHI_Score*ISI*SWLS, data=tinni_d)
summary(full_1_4)



# poly interaction
```

```
full_l_5 <- lm(TFI_Reduction ~ HHI_Score +
ISI + SWLS + I(ISI^2)+ I(HHI_Score^2)+ ISI*HHI_Score, data=tinni_d)
summary(full_l_5)
```

```
#### subset selection

l_model_R <- lm(TFI_Reduction ~ ., data=training)
summary(l_model_R)

l_model_step_AIC <- step(l_model_R,direction="both",k=2) # for AIC

summary(l_model_step_AIC)

l_model_step_BIC <- step(l_model_R,direction="both",
k=log(nrow(training))) # for BIC

summary(l_model_step_BIC)


#anova for AIC and BIC
l_model_step_AIC$anova
l_model_step_BIC$anova
```

```
# lars give k fold cross validation

library(lars)

l_model_cv<-cv.lars(x=as.matrix(training[,-12]) , y = as.matrix(training[,12]),
               K= 10, plot.it=TRUE,se=TRUE,type="stepwise")


summary(l_model_cv)




### Step 6:
# Once we select the best model with high prediction power
(using multiple metrics like
# Adjusted R2, AIC and BIC to select the best model),
#perform the model diagnostics. If we see any
# violations in the model assumptions, we will take
#the appropriate actions to correct them. (For example
#, if we see a U shape pattern in the residual plot,
try including the quadratic term, if you see any
# potential influential point, create two regression models,
```

```
both with and without that data point
# to evaluate how the regression estimates and
their standard errors get impacted).
# There will be comment on what we saw and
on the actions that we took to justify our approach.




##############
library(leaps)
#a <- regsubsets (TFI_Reduction~.,data=training,
intercept=TRUE, method = "forward")
a <- regsubsets(x=training[,-12] , y = training[,12],
                  intercept = TRUE, method = "forward")
summ_a <- summary(a)
as.data.frame(summ_a$outmat)

nad <- which.max(summ_a$adjr2)
nrss<-which.min(summ_a$rss)

ncp<-which.min(summ_a$cp)

nbic<-which.min(summ_a$bic)
```

```
# getting the model
adjr2 <- summ_a$which[nad,]

rss <- summ_a$which[nrss,]

cp <- summ_a$which[ncp,]

bic <- summ_a$which[nbic,]
cbind(adjr2,rss,cp,bic)



# coefficient of the object
coef(a,ncp) # coefficient of best model
vcov(a,ncp)

plot(a,scale="Cp")

plot(a,scale="bic")

plot(a,scale="adjr2")

plot(a,scale="r2")




### Step 7:
```

```
# After we clarify that there is no issue with the model assumptions,
#we will use the model to find
# out the factors which highly influence the reduction in TFI score.
#There will be comment on our
# findings.


# final model
#final_m <- lm(training$TFI_Reduction ~
training$HHI_Score+ training$ISI + training$PHQ + training$SWLS, data=training)
#summary(final_m)

# final model
fin <- lm(training$TFI_Reduction ~
training$HHI_Score+ training$ISI  + training$SWLS, data=training)
summary(fin)
plot(fin)

#plot(final_m)
#coef(a,ncp)


### Step 8:
# Now we will make prediction on the test data set.
#There will be comment on mean square error on the
# testing data set.

fin_test <- lm(testing$TFI_Reduction ~ testing$HHI_Score+
```

```
testing$ISI  + testing$SWLS, data=testing)
#pred <- predict(fin_test, testing[,1:11])

mse_l <- mean((fin_test$fitted.values-testing$TFI_Reduction)^2)



### Step 9:
# Now we will perform ridge regression,
#lasso regression, principal component regression, and partial least
# squares regression to do the same previous steps to decide
#which multiple regression has the highest
# predictive power via the smallest testing error



# Ridge regression  (experiment)

library(glmnet)

x <- model.matrix(TFI_Reduction ~ ., data = training) [,-1]

y <- training$TFI_Reduction

mean(y)

list.lambda <- 10^seq(1,-2,length=100)
```

```
ridge.model <- glmnet(x, y, alpha=0, lambda=list.lambda, standardize = TRUE)
#alpha =0 for ridge regression

ridge.model$lambda[10]

coef(ridge.model)[,10]  # coefficient respective to lambda [10]
dim(coef(ridge.model)[-1,])
penalty<-colSums(coef(ridge.model)[-1,])^2
plot(penalty)

predict(ridge.model,s=200, exact=T,
type="coefficients",x=scale(x),y=y) # for lambda=200

predict(ridge.model,s=ridge.model$lambda[10],
exact=T,type="coefficients",x=scale(x),y=y) # for lambda=200


predict(ridge.model,s=0, exact=T,type="coefficients",x=x,y=y) # same as original
lm(TFI_Reduction~.,data=training)$coef

## Scaling
scale.x <-scale(x)
round(colMeans(scale.x))
apply(scale.x,2,sd)


lambda<-ridge.model$lambda[10]
```

```
#### effective degree of freedom function

df<-function(lambda,matrix=x){
  svd.x<-svd(matrix)
  D<-svd.x$d
  vdfL<-c()
  for (i in 1:length(lambda)){
    dfL<-D^2/(D^2+lambda[i])
    dfL<-sum(dfL)
    vdfL<-c(vdfL,dfL)
  }
  return(vdfL)
}


df(ridge.model$lambda[10],scale.x)

plot(list.lambda,df(list.lambda))




# training ridge (main part)


 # define earlier
```

```
#train_in<-nrow(tinni_data_frame) %>% sample(.,0.8*.)

#train_in <- sample(nrow(tinni_data_frame), .8*nrow(tinni_data_frame))

#data_test <- tinni_data_frame[-train_in,]

#rownames(data_test) = c()



train_in <- train_ind
data_test <- testing
x <- model.matrix(TFI_Reduction ~ ., data = tinni_data_frame) [,-1]

y <- tinni_data_frame$TFI_Reduction


list.lambda <- 10^seq(1,-2,length=100)

#ridge.model<-glmnet(scale(x)[train_in,], y[train_in],alpha=0,lambda=list.lambda)

ridge.model<-glmnet(scale(x), y,alpha=0)

#optimal value of lambda
set.seed(123)
ridge.cv <- cv.glmnet(scale(x)[train_in,],y[train_in],alpha =0)
optimal_lam <- ridge.cv$lambda.min
```

```
optimal_second <- ridge.cv$lambda.1se

plot(ridge.cv)


#degree of freedom

df(ridge.cv$lambda.min,scale(x)[train_in,])
df(ridge.cv$lambda.1se,scale(x)[train_in,])


# prediction
#ridge.pred <-ridge.model %>% predict(s=optimal_lam, newx = x[-train_in,])

ridge.pred <-  predict(ridge.model, s=optimal_lam,
                       newx = scale(x)[-train_in,])

mse_ridge<- mean((ridge.pred - data_test$TFI_Reduction)^2)







# lasso


# only difference will alpha =1
```

```
lasso.model<-glmnet(scale(x), y,alpha=1)

#optimal value of lambda
set.seed(123)
lasso.cv <- cv.glmnet(scale(x)[train_in,],y[train_in],alpha =1)
optimal_lam <- lasso.cv$lambda.min
optimal_second <- lasso.cv$lambda.1se

plot(lasso.cv)


#degree of freedom

df(lasso.cv$lambda.min,scale(x)[train_in,])
df(lasso.cv$lambda.1se,scale(x)[train_in,])


# prediction
#ridge.pred <-ridge.model %>% predict(s=optimal_lam, newx = x[-train_in,])

lasso.pred <-  predict(lasso.model, s=optimal_lam,
                       newx = scale(x)[-train_in,])

mse_lasso<- mean((lasso.pred - data_test$TFI_Reduction)^2)
```

```
# Principal component regression

library(pls)

pcr.model <- pcr(TFI_Reduction ~ ., data=tinni_data_frame,
                 scale =TRUE, validation ="CV")  # using 10 fold
summary(pcr.model)
validationplot(pcr.model,val.type = "MSEP")




## Train data
train <- sample(1:142,80)
pcr.model <- pcr(TFI_Reduction ~ ., data= tinni_data_frame, scale =TRUE,
                 validation = "CV", subset = train)
validationplot(pcr.model, val.type = "MSEP")



# ##


pcr.model <- pcr(TFI_Reduction ~ ., data=tinni_data_frame,
                 scale =TRUE, ncomp=4)  # using 10 fold
summary(pcr.model)
```

```
pcr.model$projection
pcr.model$coefficients
pcr.model$loadings


pred.pcr<-predict(pcr.model,x[-train,],ncomp = 4)
mean((pred.pcr-y[-train])^2)

###################### PLS


pls.model<-plsr(TFI_Reduction ~ . , data = tinni_data_frame,
                scale= TRUE,valiadtion = "CV", subset = train) # 10-fold
validationplot(pls.model,val.type = "MSEP")
summary(pls.model)

pred.pls<-predict(pls.model,x[-train,],ncomp = 2)
mean((pred.pls-y[-train])^2)




######## K Means Regression #####################################



### Step 10:
```

```
# Now we will use K means regression to train
#several regression models after selecting the best k value,
# using Monte-Carlo cross validation.

#train_in<-nrow(tinni_data_frame) %>% sample(.,0.8*.)

#data_train <- tinni_data_frame[train_in,]

#data_test <- tinni_data_frame[-train_in,]

#rownames(data_test) = c()

train_in <- train_ind
data_test <- testing
data_train <-training



library(class)
library(MASS)
library(FNN)
Choose_K<-function(K,M=800){
  MSE<-0
  for(i in 1:M){
    train<-sample(113,113-23) # for 5 fold
    KNN<-knn.reg(data_train[train,1:11],
    data_train[-train,1:11], data_train[train,12], k=K)
    MSE <- MSE + mean((KNN$pred - data_train[-train, 12])^2)
  }
```

```
  CVK<-MSE/M
  return(CVK)
}

library("purrr")

LK<-1:35
vCVK<-LK %>%
  map(function(K) Choose_K(K))

vCVK<-unlist(vCVK)
vCVK<-as.data.frame(vCVK)
colnames(vCVK)<-c("CVK")

ggplot()+
  geom_point(data=vCVK,aes(x=LK,y=CVK))+
  xlab("K")+
  ylab("CV")

LK[which.min(vCVK$CVK)]



### Step 11:
# We will make some prediction on the testing data set
and obtain the mean square error.

library(class)
```

```
KNN_t <- knn.reg(data_train[,1:11],data_test[,1:11],data_train[,12],
                 k=22) # with appropriate k =29

mse_knn<- mean((KNN_t$pred - data_test$TFI_Reduction)^2)
```