MIDTERM PROJECT

1. Exploratory Data Analysis
2. What variables look most promising for predicting cancer mortality from exploratory data analysis? Why?

Ans-The variables that look promising for predicting cancer are incidenceRate, medIncome, PctHS18\_24, PctBachDeg18\_24, PctPrivateCoverage, PctPublicCoverageAlone, PctOtherRace. This can be seen by comparing the p-values that are calculated by fitting a linear regression model using the Cancer Data dataset. This can also be seen for the co-relation plot and co-relation matrix.

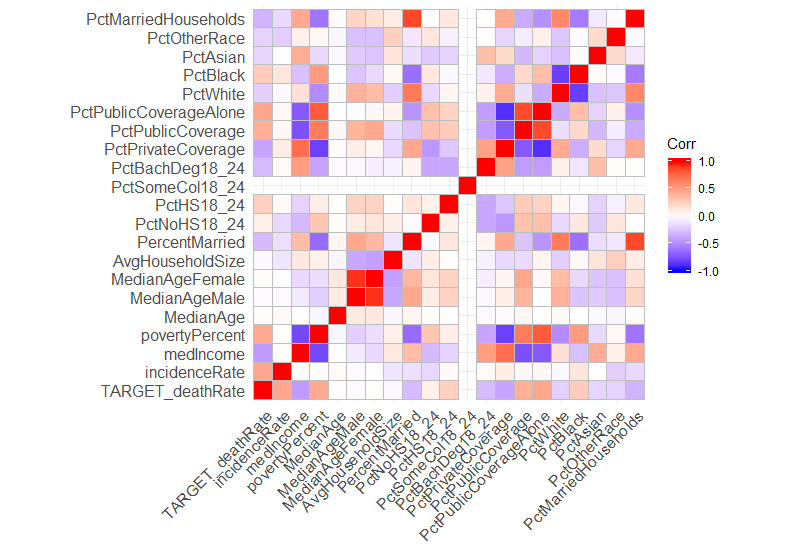


Figure 1-Co-relation graph

1. Are there any outliers? Can they be detected and addressed? How does addressing outliers affect model performance?

Ans-Yes there are outliers, this can be seen by plotting a box plot. Outliers can be treated by replacing them with the column mean or mode. They can also be treated using capping technique. In percentile capping, the value at 1st percentile, and values that are greater than the value at 99th percentile are replaced by the value at 99th percentile. In this following code, I used a for loop to replace the outliers in every feature.

Code-

#finding outliers

OutVals = boxplot(train, plot=FALSE)$out

OutVals1 = boxplot(medIncome, plot=FALSE)$out

plot(OutVals1)

plot(OutVals)

boxplot(train)

library(outliers)

outlier(medIncome)

#treating outlies- by using capping

y = c(1,2,3,4,5,6,7,9,10,11,12,14,15,16,17,18,19,20,21,22)

for (i in y)

{

x <- train[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train[,i] = x

}

for (i in y)

{

x <- test[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

test[,i] = x

}

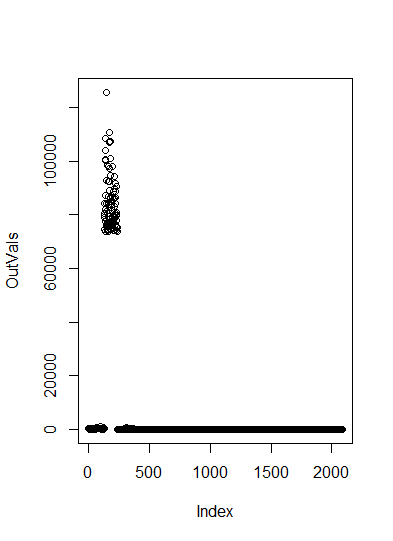


Figure 2- Outliers Plot

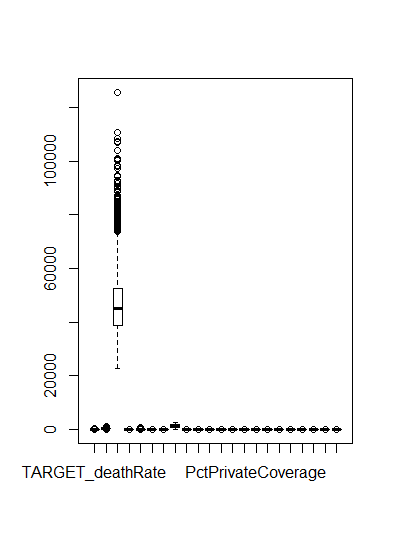


Figure 3-Box plot

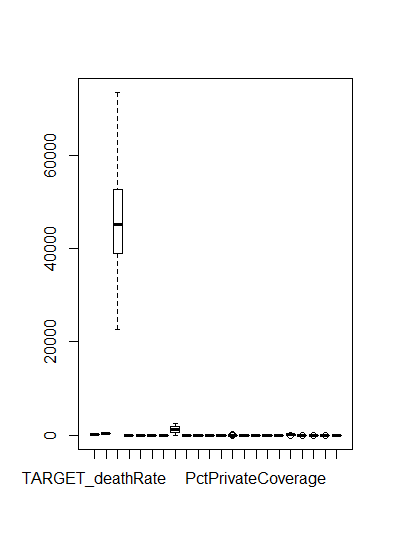


Figure 4- Box pot after capping

1. Are there any missing values? Research and explore techniques to handle missing values. Note that the approach to handle missing data might be different for different variables. Document model performance improvement obtained by missing data handling.

Ans – By observing the test and the train dataset, one can observe that there are a lot of missing values in PctSomeCol18\_24. This can also be observed by plotting a Missing Map of the datasets. There are a total of 1938 missing values in PctSomeCol18\_24. Missing values can be treated by replacing them with mean, median or mode of that column or ignoring the column if there are a lot of missing values. In this case, since there are a lot of missing values PctSomeCol18\_24 can be neglected from model fitting. Documentation of model performance improvement obtained by missing data handling is done in question 2.

Code-

#missing values

library(Amelia)

sum(is.na(train$PctSomeCol18\_24))

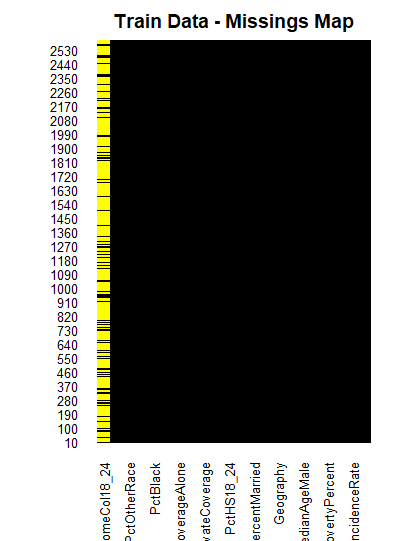
missmap(train, main="Train Data - Missings Map",

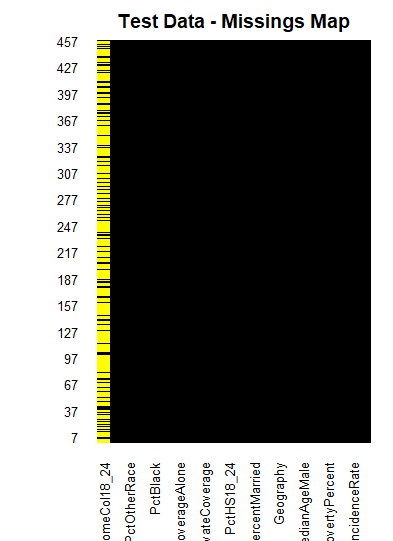
col=c("yellow", "black"), legend=FALSE)

Output-

> sum(is.na(train$PctSomeCol18\_24))

[1] 1938





1. Is there any collinearity between variables? Can it be detected? Document how addressing collinearity affects model performance?
2. Ans- There is collinearity between the variables. It can be detected by observing the VIF values of the variables after fitting them in linear regression using the olsrr library. Documentation on how addressing collinearity affects the model performance is discussed in question 2. Any variable with VIF value above 4 should be neglected because of collinearity.

Code-

#finding collinearity

install.packages('olsrr')

train = read.csv("C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerData.csv ")

test = read.csv('C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerHoldoutData.csv ')

train$PctSomeCol18\_24[is.na(train$PctSomeCol18\_24) ]= median(train$PctSomeCol18\_24, na.rm= TRUE)

test$PctSomeCol18\_24[is.na(test$PctSomeCol18\_24)]= median(test$PctSomeCol18\_24, na.rm= TRUE)

library(olsrr)

ols\_vif\_tol(LR3)

Output-

> ols\_vif\_tol(LR3)

Variables Tolerance VIF

1 incidenceRate 0.82900494 1.206265

2 medIncome 0.17124026 5.839748

3 povertyPercent 0.13454213 7.432616

4 MedianAge 0.98189629 1.018437

5 MedianAgeMale 0.11254315 8.885481

6 MedianAgeFemale 0.09976550 10.023505

7 AvgHouseholdSize 0.68497668 1.459904

8 PercentMarried 0.13710069 7.293909

9 PctNoHS18\_24 0.63776768 1.567969

10 PctHS18\_24 0.72733607 1.374880

11 PctBachDeg18\_24 0.54450499 1.836530

12 PctPrivateCoverage 0.10992638 9.096997

13 PctPublicCoverage 0.05668669 17.640825

14 PctPublicCoverageAlone 0.05545882 18.031395

15 PctWhite 0.14668757 6.817210

16 PctBlack 0.19471208 5.135788

17 PctAsian 0.57515070 1.738675

18 PctOtherRace 0.71566391 1.397304

19 PctMarriedHouseholds 0.15894287 6.291569

1. Linear Regression
2. Develop a linear regression model.

Ans- Multiple linear regression models are developed after refining the data at each step.

After treating the missing values-

* Three models are built after treating the missing by replacing them with mean, median and by neglecting the column. The train MSE obtained by neglecting the column, replacing it with median and mean is 411.3217, 411.3189, 411.3217 respectively and the test MSEs obtained are 414.5908, 414.54, 414.5908. Neglecting the column is a better choice in this situation since there are a lot of missing values even though the test MSE and train MSE are better when replaced by median.

After treating the outliers-

* The fourth model is developed after replacing the outliers using percentile capping and neglecting the PctSomeCol18\_24 column. The train and test MSE are 351.1055 and 348.365 respectively. This model performs well on the training data set since the train MSE is lower than the other three models but it has a higher test MSE compared to the other models.

After treating collinearity-

* The fifth model is developed after removing the collinear variables and neglecting the PctSomeCol18\_24 column. The train and test MSE are 459.1824 and 460.9086 respectively. This model doesn’t perform as good as the other models because it has a higher test and train MSE.

After treating everything -

* The last model is developed after treating the missing values, outliers and collinearity. The train and test MSE are 381.06 and 361.6792 respectively.

Code-

#missing values

library(Amelia)

sum(is.na(train$PctSomeCol18\_24))

missmap(train, main="Train Data - Missings Map",

col=c("yellow", "black"), legend=FALSE)

missmap(test, main="Test Data - Missings Map",

col=c("yellow", "black"), legend=FALSE)

#treating missing values

#method1 - neglecting the coloumn

LR3 = lm(TARGET\_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18\_24+PctHS18\_24+PctBachDeg18\_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

, data =train)

summary(LR3)

LR3.pred= predict(LR3 ,newdata= test )

msetrain\_n=mean((train$TARGET\_deathRate-fitted(LR3))^2)

msetrain\_n

msetest\_n=mean(((test$TARGET\_deathRate) - (LR3.pred))^2)

msetest\_n

#Method2 - inputing median

train = read.csv("C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerData.csv ")

test = read.csv('C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerHoldoutData.csv ')

train$PctSomeCol18\_24[is.na(train$PctSomeCol18\_24) ]= median(train$PctSomeCol18\_24, na.rm= TRUE)

test$PctSomeCol18\_24[is.na(test$PctSomeCol18\_24)]= median(test$PctSomeCol18\_24, na.rm= TRUE)

LR2 = lm(TARGET\_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18\_24+PctHS18\_24+PctSomeCol18\_24+PctBachDeg18\_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

, data =train)

summary(LR2)

LR2.pred= predict(LR2 ,newdata= test)

LR2.pred

msetrain\_median=mean((train$TARGET\_deathRate-fitted(LR2))^2)

msetrain\_median

msetest\_median=mean(((test$TARGET\_deathRate) - (LR2.pred))^2)

msetest\_median

#method3- Inputing the mean

train = read.csv("C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerData.csv ")

test = read.csv('C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerHoldoutData.csv ')

train$PctSomeCol18\_24[is.na(train$PctSomeCol18\_24) ]= mean(train$PctSomeCol18\_24, na.rm= TRUE)

test$PctSomeCol18\_24[is.na(test$PctSomeCol18\_24)]= mean(test$PctSomeCol18\_24, na.rm= TRUE)

LR1 = lm(TARGET\_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18\_24+PctHS18\_24+PctBachDeg18\_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

, data =train)

summary(LR1)

LR1.pred= predict(LR1 ,newdata= test)

msetrain1=mean((train$TARGET\_deathRate-fitted(LR1))^2)

msetrain1

msetest1=mean(((test$TARGET\_deathRate) - (LR1.pred))^2)

msetest1

attach(train)

#removing insignificant variables

fix(train)

LR4 = lm(TARGET\_deathRate~incidenceRate+medIncome+PctHS18\_24+PctOtherRace+PctBachDeg18\_24+PctPrivateCoverage+PctPublicCoverageAlone+povertyPercent, data =train)

LR4.pred= predict(LR4 ,newdata= test)

msetrain\_sign=mean((train$TARGET\_deathRate - fitted(LR4))^2)

msetrain\_sign

msetest\_sign=mean(((test$TARGET\_deathRate) - (LR4.pred))^2)

msetest\_sign

#finding outliers

OutVals = boxplot(train, plot=FALSE)$out

OutVals1 = boxplot(medIncome, plot=FALSE)$out

plot(OutVals1)

plot(OutVals)

boxplot(train)

library(outliers)

outlier(medIncome)

#treating outlies- by using capping

y = c(1,2,3,4,5,6,7,9,10,11,12,14,15,16,17,18,19,20,21,22)

for (i in y)

{

x <- train[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train[,i] = x

}

for (i in y)

{

x <- test[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

test[,i] = x

}

boxplot(train)

LR5 = lm(TARGET\_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18\_24+PctHS18\_24+PctBachDeg18\_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

, data =train)

summary(LR5)

LR5.pred= predict(LR5 ,newdata= test)

msetrain2=mean((train$TARGET\_deathRate-fitted(LR5))^2)

msetrain2

msetest2=mean(((test$TARGET\_deathRate) - (LR5.pred))^2)

msetest2

#finding collinearity

#install.packages('olsrr')

train = read.csv("C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerData.csv ")

test = read.csv('C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerHoldoutData.csv ')

train$PctSomeCol18\_24[is.na(train$PctSomeCol18\_24) ]= median(train$PctSomeCol18\_24, na.rm= TRUE)

test$PctSomeCol18\_24[is.na(test$PctSomeCol18\_24)]= median(test$PctSomeCol18\_24, na.rm= TRUE)

library(olsrr)

ols\_vif\_tol(LR3)

#treating collinearity - neglecting the variables

LR6 = lm(TARGET\_deathRate~incidenceRate+medIncome+MedianAge+AvgHouseholdSize+PctBlack+PctAsian+PctOtherRace, data =train)

summary(LR6)

LR6.pred= predict(LR6 ,newdata= test)

msetrain3=mean((train$TARGET\_deathRate-fitted(LR6))^2)

msetrain3

msetest3=mean(((test$TARGET\_deathRate) - (LR6.pred))^2)

msetest3

#after treating everything

train = read.csv("C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerData.csv ")

test = read.csv('C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerHoldoutData.csv ')

y = c(1,2,3,4,5,6,7,9,10,11,12,14,15,16,17,18,19,20,21,22)

for (i in y)

{

x <- train[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train[,i] = x

}

for (i in y)

{

x <- test[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

test[,i] = x

}

LR7 = lm(TARGET\_deathRate~incidenceRate+medIncome+MedianAge+AvgHouseholdSize+PctBlack+PctAsian+PctOtherRace

, data =train)

summary(LR7)

LR7.pred= predict(LR7 ,newdata= test)

msetrain4=mean((train$TARGET\_deathRate-fitted(LR7))^2)

msetrain4

msetest4=mean(((test$TARGET\_deathRate) - (LR7.pred))^2)

msetest4

Output-

> summary(LR3)

Call:

lm(formula = TARGET\_deathRate ~ incidenceRate + medIncome + povertyPercent +

MedianAge + MedianAgeMale + MedianAgeFemale + AvgHouseholdSize +

PercentMarried + PctNoHS18\_24 + PctHS18\_24 + PctBachDeg18\_24 +

PctPrivateCoverage + PctPublicCoverage + PctPublicCoverageAlone +

PctWhite + PctBlack + PctAsian + PctOtherRace + PctMarriedHouseholds,

data = train)

Residuals:

Min 1Q Median 3Q Max

-86.338 -12.160 -0.137 11.656 127.254

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 1.057e+02 1.422e+01 7.435 1.42e-13

incidenceRate 2.177e-01 8.218e-03 26.494 < 2e-16

medIncome -2.648e-04 7.983e-05 -3.317 0.000922

povertyPercent 3.093e-01 1.697e-01 1.823 0.068467

MedianAge 2.215e-03 9.630e-03 0.230 0.818095

MedianAgeMale -2.048e-01 2.292e-01 -0.893 0.371682

MedianAgeFemale -1.382e-01 2.392e-01 -0.578 0.563459

AvgHouseholdSize 6.104e-01 1.201e+00 0.508 0.611419

PercentMarried 1.748e-01 1.565e-01 1.117 0.264197

PctNoHS18\_24 -4.513e-02 6.158e-02 -0.733 0.463691

PctHS18\_24 4.582e-01 5.217e-02 8.782 < 2e-16

PctBachDeg18\_24 -3.448e-01 1.182e-01 -2.918 0.003553

PctPrivateCoverage -2.744e-01 1.135e-01 -2.417 0.015711

PctPublicCoverage 2.896e-02 2.136e-01 0.136 0.892171

PctPublicCoverageAlone 5.627e-01 2.780e-01 2.024 0.043095

PctWhite -4.835e-02 6.361e-02 -0.760 0.447280

PctBlack 3.708e-02 6.232e-02 0.595 0.551899

PctAsian -2.683e-01 1.989e-01 -1.349 0.177477

PctOtherRace -9.938e-01 1.293e-01 -7.687 2.12e-14

PctMarriedHouseholds -2.982e-01 1.531e-01 -1.947 0.051613

(Intercept) \*\*\*

incidenceRate \*\*\*

medIncome \*\*\*

povertyPercent .

MedianAge

MedianAgeMale

MedianAgeFemale

AvgHouseholdSize

PercentMarried

PctNoHS18\_24

PctHS18\_24 \*\*\*

PctBachDeg18\_24 \*\*

PctPrivateCoverage \*

PctPublicCoverage

PctPublicCoverageAlone \*

PctWhite

PctBlack

PctAsian

PctOtherRace \*\*\*

PctMarriedHouseholds .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.36 on 2570 degrees of freedom

Multiple R-squared: 0.4728, Adjusted R-squared: 0.4689

F-statistic: 121.3 on 19 and 2570 DF, p-value: < 2.2e-16

Warning messages:

1: In doTryCatch(return(expr), name, parentenv, handler) :

display list redraw incomplete

2: In doTryCatch(return(expr), name, parentenv, handler) :

invalid graphics state

3: In doTryCatch(return(expr), name, parentenv, handler) :

invalid graphics state

> summary(LR2)

Call:

lm(formula = TARGET\_deathRate ~ incidenceRate + medIncome + povertyPercent +

MedianAge + MedianAgeMale + MedianAgeFemale + AvgHouseholdSize +

PercentMarried + PctNoHS18\_24 + PctHS18\_24 + PctSomeCol18\_24 +

PctBachDeg18\_24 + PctPrivateCoverage + PctPublicCoverage +

PctPublicCoverageAlone + PctWhite + PctBlack + PctAsian +

PctOtherRace + PctMarriedHouseholds, data = train)

Residuals:

Min 1Q Median 3Q Max

-86.368 -12.179 -0.142 11.648 127.281

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 1.051e+02 1.497e+01 7.022 2.79e-12

incidenceRate 2.177e-01 8.222e-03 26.484 < 2e-16

medIncome -2.647e-04 7.985e-05 -3.314 0.000931

povertyPercent 3.094e-01 1.697e-01 1.823 0.068440

MedianAge 2.211e-03 9.632e-03 0.230 0.818497

MedianAgeMale -2.055e-01 2.293e-01 -0.896 0.370160

MedianAgeFemale -1.377e-01 2.393e-01 -0.576 0.564976

AvgHouseholdSize 6.071e-01 1.202e+00 0.505 0.613465

PercentMarried 1.758e-01 1.568e-01 1.122 0.262074

PctNoHS18\_24 -4.241e-02 6.492e-02 -0.653 0.513636

PctHS18\_24 4.610e-01 5.636e-02 8.181 4.39e-16

PctSomeCol18\_24 1.112e-02 8.397e-02 0.132 0.894639

PctBachDeg18\_24 -3.423e-01 1.197e-01 -2.860 0.004277

PctPrivateCoverage -2.747e-01 1.136e-01 -2.419 0.015653

PctPublicCoverage 2.916e-02 2.137e-01 0.136 0.891447

PctPublicCoverageAlone 5.624e-01 2.781e-01 2.022 0.043271

PctWhite -4.830e-02 6.362e-02 -0.759 0.447871

PctBlack 3.724e-02 6.234e-02 0.597 0.550318

PctAsian -2.683e-01 1.990e-01 -1.348 0.177667

PctOtherRace -9.937e-01 1.293e-01 -7.685 2.17e-14

PctMarriedHouseholds -2.991e-01 1.533e-01 -1.951 0.051200

(Intercept) \*\*\*

incidenceRate \*\*\*

medIncome \*\*\*

povertyPercent .

MedianAge

MedianAgeMale

MedianAgeFemale

AvgHouseholdSize

PercentMarried

PctNoHS18\_24

PctHS18\_24 \*\*\*

PctSomeCol18\_24

PctBachDeg18\_24 \*\*

PctPrivateCoverage \*

PctPublicCoverage

PctPublicCoverageAlone \*

PctWhite

PctBlack

PctAsian

PctOtherRace \*\*\*

PctMarriedHouseholds .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.36 on 2569 degrees of freedom

Multiple R-squared: 0.4728, Adjusted R-squared: 0.4687

F-statistic: 115.2 on 20 and 2569 DF, p-value: < 2.2e-16

> summary(LR1)

Call:

lm(formula = TARGET\_deathRate ~ incidenceRate + medIncome + povertyPercent +

MedianAge + MedianAgeMale + MedianAgeFemale + AvgHouseholdSize +

PercentMarried + PctNoHS18\_24 + PctHS18\_24 + PctBachDeg18\_24 +

PctPrivateCoverage + PctPublicCoverage + PctPublicCoverageAlone +

PctWhite + PctBlack + PctAsian + PctOtherRace + PctMarriedHouseholds,

data = train)

Residuals:

Min 1Q Median 3Q Max

-86.338 -12.160 -0.137 11.656 127.254

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 1.057e+02 1.422e+01 7.435 1.42e-13

incidenceRate 2.177e-01 8.218e-03 26.494 < 2e-16

medIncome -2.648e-04 7.983e-05 -3.317 0.000922

povertyPercent 3.093e-01 1.697e-01 1.823 0.068467

MedianAge 2.215e-03 9.630e-03 0.230 0.818095

MedianAgeMale -2.048e-01 2.292e-01 -0.893 0.371682

MedianAgeFemale -1.382e-01 2.392e-01 -0.578 0.563459

AvgHouseholdSize 6.104e-01 1.201e+00 0.508 0.611419

PercentMarried 1.748e-01 1.565e-01 1.117 0.264197

PctNoHS18\_24 -4.513e-02 6.158e-02 -0.733 0.463691

PctHS18\_24 4.582e-01 5.217e-02 8.782 < 2e-16

PctBachDeg18\_24 -3.448e-01 1.182e-01 -2.918 0.003553

PctPrivateCoverage -2.744e-01 1.135e-01 -2.417 0.015711

PctPublicCoverage 2.896e-02 2.136e-01 0.136 0.892171

PctPublicCoverageAlone 5.627e-01 2.780e-01 2.024 0.043095

PctWhite -4.835e-02 6.361e-02 -0.760 0.447280

PctBlack 3.708e-02 6.232e-02 0.595 0.551899

PctAsian -2.683e-01 1.989e-01 -1.349 0.177477

PctOtherRace -9.938e-01 1.293e-01 -7.687 2.12e-14

PctMarriedHouseholds -2.982e-01 1.531e-01 -1.947 0.051613

(Intercept) \*\*\*

incidenceRate \*\*\*

medIncome \*\*\*

povertyPercent .

MedianAge

MedianAgeMale

MedianAgeFemale

AvgHouseholdSize

PercentMarried

PctNoHS18\_24

PctHS18\_24 \*\*\*

PctBachDeg18\_24 \*\*

PctPrivateCoverage \*

PctPublicCoverage

PctPublicCoverageAlone \*

PctWhite

PctBlack

PctAsian

PctOtherRace \*\*\*

PctMarriedHouseholds .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.36 on 2570 degrees of freedom

Multiple R-squared: 0.4728, Adjusted R-squared: 0.4689

F-statistic: 121.3 on 19 and 2570 DF, p-value: < 2.2e-16

> summary(LR5)

Call:

lm(formula = TARGET\_deathRate ~ incidenceRate + medIncome + povertyPercent +

MedianAge + MedianAgeMale + MedianAgeFemale + AvgHouseholdSize +

PercentMarried + PctNoHS18\_24 + PctHS18\_24 + PctBachDeg18\_24 +

PctPrivateCoverage + PctPublicCoverage + PctPublicCoverageAlone +

PctWhite + PctBlack + PctAsian + PctOtherRace + PctMarriedHouseholds,

data = train)

Residuals:

Min 1Q Median 3Q Max

-83.779 -11.131 -0.023 11.414 65.958

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 1.323e+02 1.807e+01 7.321 3.26e-13

incidenceRate 2.170e-01 8.710e-03 24.917 < 2e-16

medIncome -4.402e-04 9.907e-05 -4.443 9.24e-06

povertyPercent -1.032e-01 1.873e-01 -0.551 0.581676

MedianAge -1.362e-01 3.321e-01 -0.410 0.681805

MedianAgeMale -2.303e-01 2.864e-01 -0.804 0.421302

MedianAgeFemale -3.389e-02 2.821e-01 -0.120 0.904381

AvgHouseholdSize 4.789e+00 3.261e+00 1.468 0.142096

PercentMarried 2.915e-02 1.632e-01 0.179 0.858288

PctNoHS18\_24 -5.393e-02 6.269e-02 -0.860 0.389773

PctHS18\_24 4.371e-01 5.080e-02 8.605 < 2e-16

PctBachDeg18\_24 -3.876e-01 1.333e-01 -2.908 0.003667

PctPrivateCoverage -4.026e-01 1.087e-01 -3.705 0.000216

PctPublicCoverage 1.773e-02 1.774e-01 0.100 0.920388

PctPublicCoverageAlone 2.791e-01 2.398e-01 1.164 0.244722

PctWhite 2.749e-02 5.928e-02 0.464 0.642868

PctBlack 1.165e-01 5.352e-02 2.176 0.029659

PctAsian -5.868e-01 3.859e-01 -1.521 0.128486

PctOtherRace -1.757e+00 1.959e-01 -8.968 < 2e-16

PctMarriedHouseholds -3.523e-01 1.778e-01 -1.982 0.047610

(Intercept) \*\*\*

incidenceRate \*\*\*

medIncome \*\*\*

povertyPercent

MedianAge

MedianAgeMale

MedianAgeFemale

AvgHouseholdSize

PercentMarried

PctNoHS18\_24

PctHS18\_24 \*\*\*

PctBachDeg18\_24 \*\*

PctPrivateCoverage \*\*\*

PctPublicCoverage

PctPublicCoverageAlone

PctWhite

PctBlack \*

PctAsian

PctOtherRace \*\*\*

PctMarriedHouseholds \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 18.81 on 2570 degrees of freedom

Multiple R-squared: 0.4695, Adjusted R-squared: 0.4656

F-statistic: 119.7 on 19 and 2570 DF, p-value: < 2.2e-16

> summary(LR6)

Call:

lm(formula = TARGET\_deathRate ~ incidenceRate + medIncome + MedianAge +

AvgHouseholdSize + PctBlack + PctAsian + PctOtherRace, data = train)

Residuals:

Min 1Q Median 3Q Max

-79.812 -13.020 -0.769 12.190 122.468

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 1.074e+02 4.924e+00 21.813 < 2e-16 \*\*\*

incidenceRate 2.263e-01 8.180e-03 27.666 < 2e-16 \*\*\*

medIncome -9.349e-04 4.053e-05 -23.067 < 2e-16 \*\*\*

MedianAge -2.352e-03 1.008e-02 -0.233 0.815

AvgHouseholdSize 5.421e+00 1.096e+00 4.948 7.98e-07 \*\*\*

PctBlack 2.004e-01 3.082e-02 6.504 9.33e-11 \*\*\*

PctAsian -1.709e-02 1.804e-01 -0.095 0.925

PctOtherRace -6.210e-01 1.231e-01 -5.044 4.87e-07 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 21.46 on 2582 degrees of freedom

Multiple R-squared: 0.4115, Adjusted R-squared: 0.4099

F-statistic: 257.9 on 7 and 2582 DF, p-value: < 2.2e-16

> summary(LR7)

Call:

lm(formula = TARGET\_deathRate ~ incidenceRate + medIncome + MedianAge +

AvgHouseholdSize + PctBlack + PctAsian + PctOtherRace, data = train)

Residuals:

Min 1Q Median 3Q Max

-82.077 -11.678 -0.145 11.731 73.401

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 8.689e+01 9.921e+00 8.758 < 2e-16 \*\*\*

incidenceRate 2.284e-01 8.602e-03 26.548 < 2e-16 \*\*\*

medIncome -1.006e-03 4.437e-05 -22.683 < 2e-16 \*\*\*

MedianAge -6.065e-02 1.044e-01 -0.581 0.56134

AvgHouseholdSize 1.610e+01 2.363e+00 6.815 1.17e-11 \*\*\*

PctBlack 1.651e-01 3.080e-02 5.361 9.01e-08 \*\*\*

PctAsian -1.086e+00 3.556e-01 -3.055 0.00228 \*\*

PctOtherRace -1.350e+00 1.911e-01 -7.062 2.10e-12 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 19.55 on 2582 degrees of freedom

Multiple R-squared: 0.4242, Adjusted R-squared: 0.4226

F-statistic: 271.8 on 7 and 2582 DF, p-value: < 2.2e-16

msetrain\_n

[1] 411.3217

> msetest\_n

[1] 414.5908

> msetrain\_median

[1] 411.3189

> msetest\_median

[1] 414.54

> msetrain1

[1] 411.3217

> msetest1

[1] 414.5908

> msetrain2

[1] 351.1055

> msetest2

[1] 348.3685

> msetrain3

[1] 459.1824

> msetest3

[1] 460.9086

> msetrain4 #optimum msetrain

[1] 381.0683

> msetest4 #optimum msetest

[1] 361.6792

1. What variables are significant? Insignificant? How does removing insignificant variables affect model performance?

Ans- The variables incidenceRate, medIncome, PctHS18\_24, PctBachDeg18\_24, PctPrivateCoverage, PctPublicCoverageAlone, PctOtherRace are significant.

The model performs better than the original model because it has a lower test and train MSE compared to the original model .

Code-

#removing insignificant variables

fix(train)

LR4 = lm(TARGET\_deathRate~incidenceRate+medIncome+PctHS18\_24+PctOtherRace+PctBachDeg18\_24+PctPrivateCoverage+PctPublicCoverageAlone+povertyPercent, data =train)

summary(LR4)

LR4.pred= predict(LR4 ,newdata= test)

msetrain\_sign=mean((train$TARGET\_deathRate - fitted(LR4))^2)

msetrain\_sign

msetest\_sign=mean(((test$TARGET\_deathRate) - (LR4.pred))^2)

msetest\_sign

Output-

> LR4 = lm(TARGET\_deathRate~incidenceRate+medIncome+PctHS18\_24+PctOtherRace+PctBachDeg18\_24+PctPrivateCoverage+PctPublicCoverageAlone+povertyPercent, data =train)

> summary(LR4)

Call:

lm(formula = TARGET\_deathRate ~ incidenceRate + medIncome + PctHS18\_24 +

PctOtherRace + PctBachDeg18\_24 + PctPrivateCoverage + PctPublicCoverageAlone +

povertyPercent, data = train)

Residuals:

Min 1Q Median 3Q Max

-85.564 -11.515 0.276 11.721 67.194

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 9.595e+01 1.034e+01 9.281 < 2e-16

incidenceRate 2.292e-01 8.517e-03 26.907 < 2e-16

medIncome -2.451e-04 7.516e-05 -3.261 0.00113

PctHS18\_24 3.882e-01 4.899e-02 7.925 3.38e-15

PctOtherRace -1.625e+00 1.823e-01 -8.915 < 2e-16

PctBachDeg18\_24 -3.301e-01 1.239e-01 -2.664 0.00777

PctPrivateCoverage -4.743e-01 9.692e-02 -4.893 1.05e-06

PctPublicCoverageAlone 1.762e-01 1.492e-01 1.181 0.23782

povertyPercent 5.723e-01 1.463e-01 3.912 9.38e-05

(Intercept) \*\*\*

incidenceRate \*\*\*

medIncome \*\*

PctHS18\_24 \*\*\*

PctOtherRace \*\*\*

PctBachDeg18\_24 \*\*

PctPrivateCoverage \*\*\*

PctPublicCoverageAlone

povertyPercent \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 18.98 on 2581 degrees of freedom

Multiple R-squared: 0.4577, Adjusted R-squared: 0.456

F-statistic: 272.3 on 8 and 2581 DF, p-value: < 2.2e-16

|  |
| --- |
| > msetrain\_sign  [1] 358.9149  > msetest\_sign=mean(((test$TARGET\_deathRate) - (LR4.pred))^2)  > msetest\_sign  [1] 352.1031 |
|  |
| |  | | --- | | > | |

1. Present and interpret model diagnosis. What insights did you obtain to improve the model from diagnosis?

Ans-

Code-

# model diagnosis

par(mfrow=c(2,2))

plot(LR3)

Output-

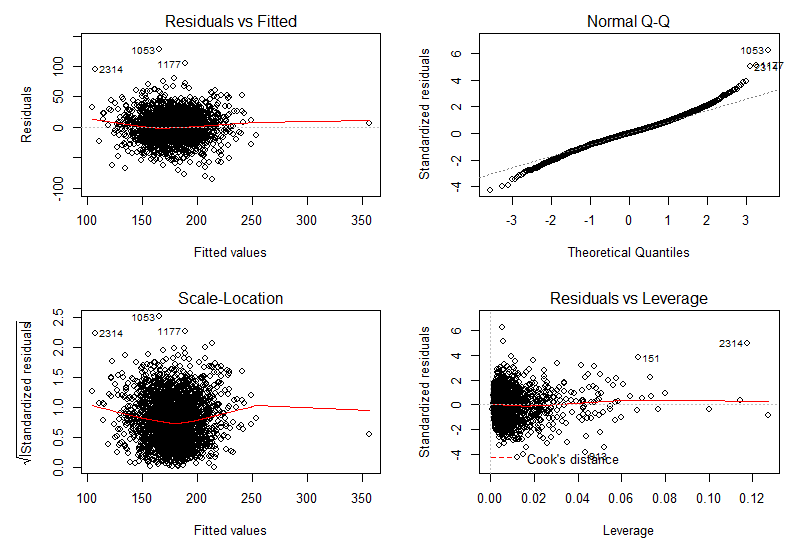


Figure 6-Diagnostics Plot

1. Plot-1 (Residual vs Fitted)- - This plot is used to check the linear relationship assumptions (This plot shows if residuals have non-linear patterns). From the diagnostic plot drawn, the residuals have a linear relationship.
2. Plot-2 (Normal Q-Q)- This plot shows if the residuals are normally distributed. They are normally distributed if all the points fall on a straight line. From the plot obtained, the residuals are normally distributed.
3. Plot-3 (Scale- Location)- It’s also called Spread-Location plot. This plot shows if residuals are spread equally along the ranges of predictors (homogeneity of variance of the residuals). From the plot obtained the residuals are spread equally along the range of the predictors.
4. Plot-4(Residuals vs leverage)- This plot helps us to find if the outliers are influential in linear regression analysis. This can be found out by Cook’s distance. From our plot we can infer that the outliers are not influential since we can’t see cook’s distance line (since its well inside the Cook’s distance line).
5. Include few non-linear and interaction terms and evaluate how they affect model performance and diagnosis.

Ans-

Code-

#inputing non-linear terms

attach(train)

LR8 = lm(TARGET\_deathRate~incidenceRate+sqrt(medIncome)+povertyPercent+MedianAge+sqrt(MedianAgeMale)+MedianAgeFemale+AvgHouseholdSize+(PercentMarried)^2+PctNoHS18\_24^3+PctHS18\_24+PctBachDeg18\_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

:medIncome, data =train)

summary(LR8)

LR8.pred= predict(LR8 ,newdata= test)

msetrain5=mean((train$TARGET\_deathRate-fitted(LR8))^2)

msetrain5 #optimum msetrain

msetest5=mean(((test$TARGET\_deathRate) - (LR8.pred))^2)

msetest5 #optimum msetest

par(mfrow=c(2,2))

plot(LR8)

Output-

|  |
| --- |
| summary(LR8)  Call:  lm(formula = TARGET\_deathRate ~ incidenceRate + sqrt(medIncome) +  povertyPercent + MedianAge + sqrt(MedianAgeMale) + MedianAgeFemale +  AvgHouseholdSize + (PercentMarried)^2 + PctNoHS18\_24^3 +  PctHS18\_24 + PctBachDeg18\_24 + PctPrivateCoverage + PctPublicCoverage +  PctPublicCoverageAlone + PctWhite + PctBlack + PctAsian +  PctOtherRace + PctMarriedHouseholds:medIncome, data = train)  Residuals:  Min 1Q Median 3Q Max  -82.478 -11.177 -0.031 11.233 64.401  Coefficients:  Estimate Std. Error t value  (Intercept) 1.692e+02 3.031e+01 5.583  incidenceRate 2.195e-01 8.667e-03 25.324  sqrt(medIncome) -2.316e-01 8.678e-02 -2.669  povertyPercent -2.853e-01 1.961e-01 -1.455  MedianAge -2.739e-01 3.293e-01 -0.832  sqrt(MedianAgeMale) 8.192e-01 3.599e+00 0.228  MedianAgeFemale -1.785e-01 2.826e-01 -0.632  AvgHouseholdSize 1.866e+00 3.171e+00 0.588  PercentMarried -2.252e-01 1.448e-01 -1.555  PctNoHS18\_24 -5.272e-02 6.269e-02 -0.841  PctHS18\_24 4.306e-01 5.080e-02 8.476  PctBachDeg18\_24 -3.674e-01 1.319e-01 -2.785  PctPrivateCoverage -4.055e-01 1.084e-01 -3.741  PctPublicCoverage -3.973e-02 1.773e-01 -0.224  PctPublicCoverageAlone 3.303e-01 2.397e-01 1.378  PctWhite -3.537e-03 5.789e-02 -0.061  PctBlack 1.048e-01 5.354e-02 1.958  PctAsian -5.315e-01 3.844e-01 -1.383  PctOtherRace -1.705e+00 1.975e-01 -8.633  PctMarriedHouseholds:medIncome -8.017e-07 2.829e-06 -0.283  Pr(>|t|)  (Intercept) 2.62e-08 \*\*\*  incidenceRate < 2e-16 \*\*\*  sqrt(medIncome) 0.007653 \*\*  povertyPercent 0.145806  MedianAge 0.405539  sqrt(MedianAgeMale) 0.819968  MedianAgeFemale 0.527553  AvgHouseholdSize 0.556344  PercentMarried 0.120070  PctNoHS18\_24 0.400444  PctHS18\_24 < 2e-16 \*\*\*  PctBachDeg18\_24 0.005395 \*\*  PctPrivateCoverage 0.000187 \*\*\*  PctPublicCoverage 0.822709  PctPublicCoverageAlone 0.168301  PctWhite 0.951277  PctBlack 0.050344 .  PctAsian 0.166922  PctOtherRace < 2e-16 \*\*\*  PctMarriedHouseholds:medIncome 0.776871  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 18.8 on 2570 degrees of freedom  Multiple R-squared: 0.47, Adjusted R-squared: 0.4661  F-statistic: 120 on 19 and 2570 DF, p-value: < 2.2e-16  > LR8.pred= predict(LR8 ,newdata= test)  > msetrain5=mean((train$TARGET\_deathRate-fitted(LR8))^2)  > msetrain5 #optimum msetrain  [1] 350.7415  > msetest5=mean(((test$TARGET\_deathRate) - (LR8.pred))^2)  > msetest5 #optimum msetest  [1] 346.4466 |
|  |
| |  | | --- | | > | |

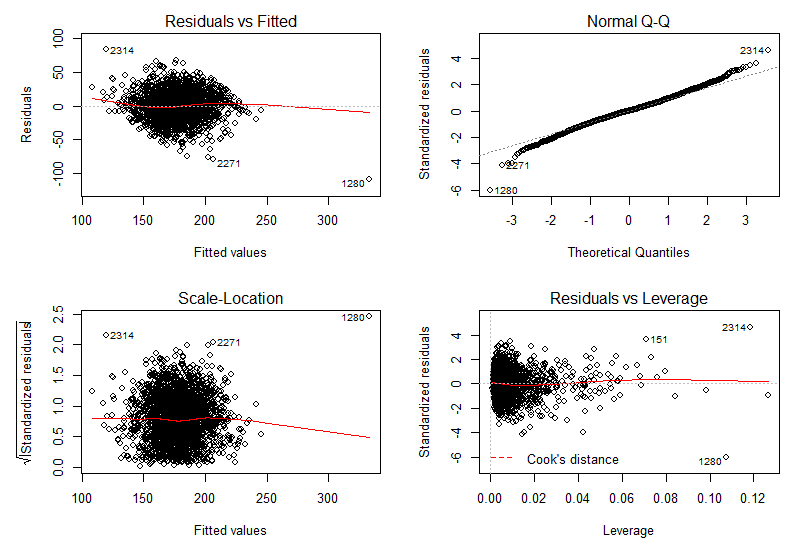


Figure 7-Diagnostics Plot

Summary- The model performance increases due to the addition of interaction terms and non-linear terms since the test and train MSE (350.74 and 346.446 respectively) are lower when compared to the test and train MSE of the original model.

1.Plot-1 (Residual vs Fitted)- - This plot is used to check the linear relationship assumptions (This plot shows if residuals have non-linear patterns). From the diagnostic plot drawn, the residuals have a linear relationship.

2.Plot-2 (Normal Q-Q)- This plot shows if the residuals are normally distributed. They are normally distributed if all the points fall on a straight line. From the plot obtained, the residuals are normally distributed.

3.Plot-3 (Scale- Location)- It’s also called Spread-Location plot. This plot shows if residuals are spread equally along the ranges of predictors (homogeneity of variance of the residuals). From the plot obtained the residuals are spread equally along the range of the predictors.

4.Plot-4(Residuals vs leverage)- This plot helps us to find if the outliers are influential in linear regression analysis. This can be found out by Cook’s distance. From our plot we can infer that the outliers are not influential since we can’t see cook’s distance line (since its well inside the Cook’s distance line).

1. KNN-
2. Split CanverData.csv data into 70% training and 30% testing.

Code-

#question 3

library(FNN)

library(class)

set.seed(1)

train = read.csv("C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerData.csv ")

test = read.csv('C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerHoldoutData.csv ')

for (i in y)

{

x <- train[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train[,i] = x

}

for (i in y)

{

x <- test[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

test[,i] = x

}

n <- nrow(train) \* 0.7

T <- sample(nrow(train), size = n)

train1 <- train[T,-c(1,8,13)]

test1 <- train[-T,-c(1,8,13)]

test1\_full <- train[-T,]

train.Y = train$TARGET\_deathRate

#fix(train1)

1. Develop KNN model for predicting Cancer Mortality. Evaluate test MSE for at least 5 different values of K and find the K that minimizes test MSE.

Ans- KNN model is developed after splitting the train into test and train data.

The K that minimizes test MSE is K=5, with a test MSE of 764.3161

Code-

knn <- knn.reg(train1, test1, train.Y, k=1)

knntestmse =mean(((test1\_full$TARGET\_deathRate) - (knn$pred))^2)

error = c(0,0,0,0,0)

for(i in 1:5)

{

knn <- knn.reg(train1, test1, train.Y, k=i)

knntestmse =mean(((test1\_full$TARGET\_deathRate) - (knn$pred))^2)

error[i] = knntestmse

}

error

Output-

> error

[1] 1259.7968 965.2434 836.5874 807.6674 764.3161

1. KNN is a non-linear technique, but does not work well with high dimensional data. Try to identify important variables from Linear Regression model and use only a subset of important features in the KNN model. Document impact on test performance.

Ans- The variables incidenceRate, medIncome, PctHS18\_24, PctBachDeg18\_24, PctPrivateCoverage, PctPublicCoverageAlone, PctOtherRace are significant. This can be found out using the p-values obtained from the linear regression. The test MSE for K = 1,2,3,4,5 is 1299.8198, 961.2180, 844.4425, 805.0395 and 760.8135. Using significant variables improved the performance of the KNN model since the test MSE for the same seed is less when compared to the KNN model when all the variables are used. K=5 is the optimum K values since the test MSE is 760.8135.

Code-

train2 <- train[T,-c(1,4,5,7,8,9,10,12,13,15,17,22)]

test2 <- train[-T,-c(1,4,5,7,8,9,10,12,13,15,17,22)]

test2\_full<-train[-T,]

train.Y1 = train$TARGET\_deathRate

#fix(train2)

knn3 <- knn.reg(train2, test2, train.Y1, k=1)

knntestmse3 =mean(((test2\_full$TARGET\_deathRate) - (knn3$pred))^2)

error2 = c(0,0,0,0,0)

for(i in 1:5)

{

knn3 <- knn.reg(train2, test2, train.Y1,k=i)

knntestmse3 =mean(((test2\_full$TARGET\_deathRate) - (knn3$pred))^2)

error2[i] = knntestmse3

}

error2

Output-

error2

[1] 1299.8198 961.2180 844.4425 805.0395 760.8135

1. Feature Selection
2. Write an “Executive Summary” section documenting your interpretation of the important features impacting cancer mortality and how they influence cancer mortality.

Ans – The feature selection from a data set in R can be done creating a correlation

matrix. Visually it can also be done by plotting a correlation plot from the matrix.

In this project correlation matrix and correlation plot are used to select the features in the initial stages. Later the p-values obtained from the linear regression are used to

select the significant features that are used to improve the model performance.

Interpreting Correlation plot-

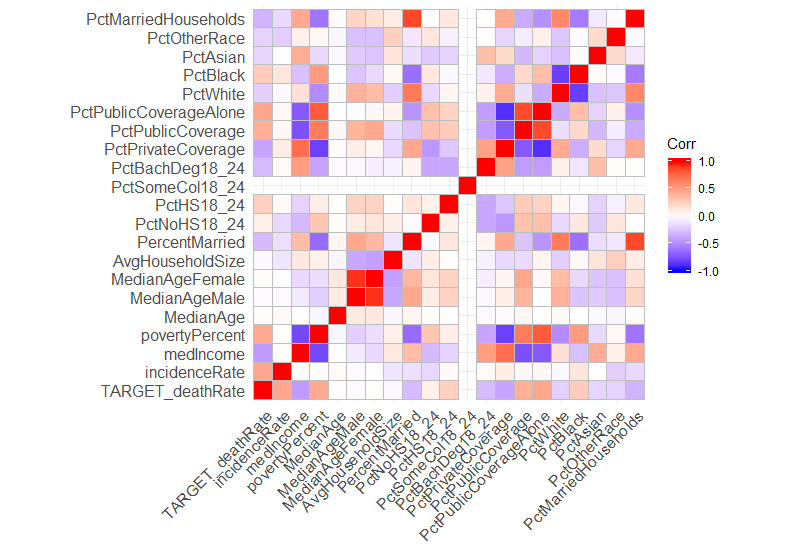


Figure 8-Correlation graph

In this correlation plot, correlation scale which is displayed in the right hand corner of the plot is used to select a feature( darker the color, the higher the correlation).

Interpreting Linear Regression-

> summary(LR3)

Call:

lm(formula = TARGET\_deathRate ~ incidenceRate + medIncome + povertyPercent +

MedianAge + MedianAgeMale + MedianAgeFemale + AvgHouseholdSize +

PercentMarried + PctNoHS18\_24 + PctHS18\_24 + PctBachDeg18\_24 +

PctPrivateCoverage + PctPublicCoverage + PctPublicCoverageAlone +

PctWhite + PctBlack + PctAsian + PctOtherRace + PctMarriedHouseholds,

data = train)

Residuals:

Min 1Q Median 3Q Max

-86.338 -12.160 -0.137 11.656 127.254

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 1.057e+02 1.422e+01 7.435 1.42e-13

incidenceRate 2.177e-01 8.218e-03 26.494 < 2e-16

medIncome -2.648e-04 7.983e-05 -3.317 0.000922

povertyPercent 3.093e-01 1.697e-01 1.823 0.068467

MedianAge 2.215e-03 9.630e-03 0.230 0.818095

MedianAgeMale -2.048e-01 2.292e-01 -0.893 0.371682

MedianAgeFemale -1.382e-01 2.392e-01 -0.578 0.563459

AvgHouseholdSize 6.104e-01 1.201e+00 0.508 0.611419

PercentMarried 1.748e-01 1.565e-01 1.117 0.264197

PctNoHS18\_24 -4.513e-02 6.158e-02 -0.733 0.463691

PctHS18\_24 4.582e-01 5.217e-02 8.782 < 2e-16

PctBachDeg18\_24 -3.448e-01 1.182e-01 -2.918 0.003553

PctPrivateCoverage -2.744e-01 1.135e-01 -2.417 0.015711

PctPublicCoverage 2.896e-02 2.136e-01 0.136 0.892171

PctPublicCoverageAlone 5.627e-01 2.780e-01 2.024 0.043095

PctWhite -4.835e-02 6.361e-02 -0.760 0.447280

PctBlack 3.708e-02 6.232e-02 0.595 0.551899

PctAsian -2.683e-01 1.989e-01 -1.349 0.177477

PctOtherRace -9.938e-01 1.293e-01 -7.687 2.12e-14

PctMarriedHouseholds -2.982e-01 1.531e-01 -1.947 0.051613

(Intercept) \*\*\*

incidenceRate \*\*\*

medIncome \*\*\*

povertyPercent .

MedianAge

MedianAgeMale

MedianAgeFemale

AvgHouseholdSize

PercentMarried

PctNoHS18\_24

PctHS18\_24 \*\*\*

PctBachDeg18\_24 \*\*

PctPrivateCoverage \*

PctPublicCoverage

PctPublicCoverageAlone \*

PctWhite

PctBlack

PctAsian

PctOtherRace \*\*\*

PctMarriedHouseholds .

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Features can be selected from linear regression by observing the corresponding p-values. Smaller the p-value, higher the significance. Significance code displayed at the bottom can be used to interpret the p-value.

1. Performance reporting on Holdout data-
2. Summarize and compare the model performance (MSE) of LR and KNN on holdout dataset as a table.

Code-

#question-5

set.seed(1)

train = read.csv("C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerData.csv ")

test = read.csv('C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerHoldoutData.csv ')

for (i in y)

{

x <- train[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train[,i] = x

}

for (i in y)

{

x <- test[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

test[,i] = x

}

trainn <- train[,-c(1,8,13)]

testn <- test[,-c(1,8,13)]

y = train$TARGET\_deathRate

error1 = c(0,0,0,0,0)

for(i in 1:5)

{

knn1<- knn.reg(trainn, testn, train$TARGET\_deathRate, k=i)

knntestmse1 =mean(((test$TARGET\_deathRate) - (knn1$pred))^2)

error1[i] = knntestmse1

}

error1

Output-

> error1

[1] 732.3953 589.5190 524.3679 514.2016 516.9247

|  |  |  |
| --- | --- | --- |
| SR. No | Model name | Test MSE |
| 1 | Linear Regression | 414.5908 |
| 3 | KNN | 514.2016 |

Summary-

The Test MSE for Linear Regression and KNN are 414.5908, 514.2016 respectively. Linear Regression perform better than KNN since KNN doesn’t work when multi-dimensional data and also when the variables are linearly separable.

R-CODE(FULL)-

#question 2

#promising variables

train = read.csv("C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerData.csv ")

test = read.csv('C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerHoldoutData.csv ')

library(ggplot2)

mydata <- train[, -c(8)]

cormat<-signif(cor(mydata),2)

cormat

install.packages("ggcorrplot")

library(ggcorrplot)

ggcorrplot(cormat)

#missing values

library(Amelia)

sum(is.na(train$PctSomeCol18\_24))

missmap(train, main="Train Data - Missings Map",

col=c("yellow", "black"), legend=FALSE)

missmap(test, main="Test Data - Missings Map",

col=c("yellow", "black"), legend=FALSE)

#treating missing values

#method1 - neglecting the coloumn

LR3 = lm(TARGET\_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18\_24+PctHS18\_24+PctBachDeg18\_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

, data =train)

summary(LR3)

LR3.pred= predict(LR3 ,newdata= test )

msetrain\_n=mean((train$TARGET\_deathRate-fitted(LR3))^2)

msetrain\_n

msetest\_n=mean(((test$TARGET\_deathRate) - (LR3.pred))^2)

msetest\_n

#Method2 - inputing median

train = read.csv("C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerData.csv ")

test = read.csv('C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerHoldoutData.csv ')

train$PctSomeCol18\_24[is.na(train$PctSomeCol18\_24) ]= median(train$PctSomeCol18\_24, na.rm= TRUE)

test$PctSomeCol18\_24[is.na(test$PctSomeCol18\_24)]= median(test$PctSomeCol18\_24, na.rm= TRUE)

LR2 = lm(TARGET\_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18\_24+PctHS18\_24+PctSomeCol18\_24+PctBachDeg18\_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

, data =train)

summary(LR2)

LR2.pred= predict(LR2 ,newdata= test)

LR2.pred

msetrain\_median=mean((train$TARGET\_deathRate-fitted(LR2))^2)

msetrain\_median

msetest\_median=mean(((test$TARGET\_deathRate) - (LR2.pred))^2)

msetest\_median

#method3- Inputing the mean

train = read.csv("C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerData.csv ")

test = read.csv('C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerHoldoutData.csv ')

train$PctSomeCol18\_24[is.na(train$PctSomeCol18\_24) ]= mean(train$PctSomeCol18\_24, na.rm= TRUE)

test$PctSomeCol18\_24[is.na(test$PctSomeCol18\_24)]= mean(test$PctSomeCol18\_24, na.rm= TRUE)

LR1 = lm(TARGET\_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18\_24+PctHS18\_24+PctBachDeg18\_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

, data =train)

summary(LR1)

LR1.pred= predict(LR1 ,newdata= test)

msetrain1=mean((train$TARGET\_deathRate-fitted(LR1))^2)

msetrain1

msetest1=mean(((test$TARGET\_deathRate) - (LR1.pred))^2)

msetest1

#finding outliers

OutVals = boxplot(train, plot=FALSE)$out

OutVals1 = boxplot(medIncome, plot=FALSE)$out

plot(OutVals1)

plot(OutVals)

boxplot(train)

library(outliers)

outlier(medIncome)

#treating outlies- by using capping

y = c(1,2,3,4,5,6,7,9,10,11,12,14,15,16,17,18,19,20,21,22)

for (i in y)

{

x <- train[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train[,i] = x

}

for (i in y)

{

x <- test[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

test[,i] = x

}

boxplot(train)

LR5 = lm(TARGET\_deathRate~incidenceRate+medIncome+povertyPercent+MedianAge+MedianAgeMale+MedianAgeFemale+AvgHouseholdSize+PercentMarried+PctNoHS18\_24+PctHS18\_24+PctBachDeg18\_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

, data =train)

summary(LR5)

LR5.pred= predict(LR5 ,newdata= test)

msetrain2=mean((train$TARGET\_deathRate-fitted(LR5))^2)

msetrain2

msetest2=mean(((test$TARGET\_deathRate) - (LR5.pred))^2)

msetest2

#finding collinearity

#install.packages('olsrr')

train = read.csv("C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerData.csv ")

test = read.csv('C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerHoldoutData.csv ')

train$PctSomeCol18\_24[is.na(train$PctSomeCol18\_24) ]= median(train$PctSomeCol18\_24, na.rm= TRUE)

test$PctSomeCol18\_24[is.na(test$PctSomeCol18\_24)]= median(test$PctSomeCol18\_24, na.rm= TRUE)

library(olsrr)

ols\_vif\_tol(LR3)

#treating collinearity - neglecting the variables

LR6 = lm(TARGET\_deathRate~incidenceRate+medIncome+MedianAge+AvgHouseholdSize+PctBlack+PctAsian+PctOtherRace, data =train)

summary(LR6)

LR6.pred= predict(LR6 ,newdata= test)

msetrain3=mean((train$TARGET\_deathRate-fitted(LR6))^2)

msetrain3

msetest3=mean(((test$TARGET\_deathRate) - (LR6.pred))^2)

msetest3

#after treating everything

train = read.csv("C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerData.csv ")

test = read.csv('C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerHoldoutData.csv ')

y = c(1,2,3,4,5,6,7,9,10,11,12,14,15,16,17,18,19,20,21,22)

for (i in y)

{

x <- train[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train[,i] = x

}

for (i in y)

{

x <- test[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

test[,i] = x

}

LR7 = lm(TARGET\_deathRate~incidenceRate+medIncome+MedianAge+AvgHouseholdSize+PctBlack+PctAsian+PctOtherRace

, data =train)

summary(LR7)

LR7.pred= predict(LR7 ,newdata= test)

msetrain4=mean((train$TARGET\_deathRate-fitted(LR7))^2)

msetrain4

msetest4=mean(((test$TARGET\_deathRate) - (LR7.pred))^2)

msetest4

#removing insignificant variables

fix(train)

LR4 = lm(TARGET\_deathRate~incidenceRate+medIncome+PctHS18\_24+PctOtherRace+PctBachDeg18\_24+PctPrivateCoverage+PctPublicCoverageAlone+povertyPercent, data =train)

summary(LR4)

LR4.pred= predict(LR4 ,newdata= test)

msetrain\_sign=mean((train$TARGET\_deathRate - fitted(LR4))^2)

msetrain\_sign

msetest\_sign=mean(((test$TARGET\_deathRate) - (LR4.pred))^2)

msetest\_sign

#inputing non-linear terms

attach(train)

LR8 = lm(TARGET\_deathRate~incidenceRate+sqrt(medIncome)+povertyPercent+MedianAge+sqrt(MedianAgeMale)+MedianAgeFemale+AvgHouseholdSize+(PercentMarried)^2+PctNoHS18\_24^3+PctHS18\_24+PctBachDeg18\_24+PctPrivateCoverage+PctPublicCoverage+PctPublicCoverageAlone+PctWhite+PctBlack+PctAsian+PctOtherRace+PctMarriedHouseholds

:medIncome, data =train)

summary(LR8)

LR8.pred= predict(LR8 ,newdata= test)

msetrain5=mean((train$TARGET\_deathRate-fitted(LR8))^2)

msetrain5

msetest5=mean(((test$TARGET\_deathRate) - (LR8.pred))^2)

msetest5

par(mfrow=c(2,2))

plot(LR8)

# model diagnosis

par(mfrow=c(2,2))

plot(LR1)

#trainmse vs testmse

trainMSE= c(459,411,409,409)

testMSE= c(460,414,416,416)

#1= collinearity,2= neglecting, 3= optimum in x, 4= outliers,

x= c(1,2,3,4)

plot(x,trainMSE, ylab='trainMSE and testMSE')

lines(testMSE, col = 'red')

lines(trainMSE, col='blue')

#question 3

library(FNN)

library(class)

set.seed(1)

train = read.csv("C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerData.csv ")

test = read.csv('C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerHoldoutData.csv ')

for (i in y)

{

x <- train[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train[,i] = x

}

for (i in y)

{

x <- test[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

test[,i] = x

}

n <- nrow(train) \* 0.7

T <- sample(nrow(train), size = n)

train1 <- train[T,-c(1,8,13)]

test1 <- train[-T,-c(1,8,13)]

test1\_full <- train[-T,]

train.Y = train$TARGET\_deathRate

#fix(train1)

knn <- knn.reg(train1, test1, train.Y, k=1)

knntestmse =mean(((test1\_full$TARGET\_deathRate) - (knn$pred))^2)

error = c(0,0,0,0,0)

for(i in 1:5)

{

knn <- knn.reg(train1, test1, train.Y, k=i)

knntestmse =mean(((test1\_full$TARGET\_deathRate) - (knn$pred))^2)

error[i] = knntestmse

}

error

train2 <- train[T,-c(1,4,5,7,8,9,10,12,13,15,17,22)]

test2 <- train[-T,-c(1,4,5,7,8,9,10,12,13,15,17,22)]

test2\_full<-train[-T,]

train.Y1 = train$TARGET\_deathRate

#fix(train2)

knn3 <- knn.reg(train2, test2, train.Y1, k=1)

knntestmse3 =mean(((test2\_full$TARGET\_deathRate) - (knn3$pred))^2)

error2 = c(0,0,0,0,0)

for(i in 1:5)

{

knn3 <- knn.reg(train2, test2, train.Y1,k=i)

knntestmse3 =mean(((test2\_full$TARGET\_deathRate) - (knn3$pred))^2)

error2[i] = knntestmse3

}

error2

#question-5

set.seed(1)

train = read.csv("C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerData.csv ")

test = read.csv('C:\\Users\\KIRAN KONDISETTI\\Desktop\\CancerHoldoutData.csv ')

for (i in y)

{

x <- train[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

train[,i] = x

}

for (i in y)

{

x <- test[,i]

qnt <- quantile(x, probs=c(.25, .75))

caps <- quantile(x, probs=c(.05, .95))

H <- 1.5 \* IQR(x)

x[x < (qnt[1] - H)] <- caps[1]

x[x > (qnt[2] + H)] <- caps[2]

test[,i] = x

}

trainn <- train[,-c(1,8,13)]

testn <- test[,-c(1,8,13)]

y = train$TARGET\_deathRate

error1 = c(0,0,0,0,0)

for(i in 1:5)

{

knn1<- knn.reg(trainn, testn, train$TARGET\_deathRate, k=i)

knntestmse1 =mean(((test$TARGET\_deathRate) - (knn1$pred))^2)

error1[i] = knntestmse1

}

error1