

**UNSUPERVISED LEARNING- REDUCING MUTLI-COLLINEARITY  
,OVERFITTING AND LINERAIZING THE DATA**

**BOX-COX,RIDGE,PCA AND PARTIAL LEAST SQUARE**

**AMANPREET SINGH  
942091**

# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>  | <b>1</b>  |
| 1.1      | OVERFITTING . . . . .  | 1         |
| 1.2      | BOX-COX . . . . .  | 1         |
| 1.3      | PRINCIPAL COMPONENT ANALYSIS . . . . .   | 1         |
| 1.4      | Partial Least Square . . . . .   | 2         |
| 1.5      | Variance Inflation Factor . . . . .  | 2         |
| <b>2</b> | <b>REDUCING MUTLI-COLLINEARITY ,OVERFITTING AND LINERAIZING with R</b>   | <b>3</b>  |
| 2.1      | Call Housing Data . . . . .  | 3         |
| <b>3</b> | <b>Research</b>  | <b>4</b>  |
| 3.1      | Scatter-plot matrix to see Linearity among Variables . . . . .   | 4         |
| 3.1.1    | Splitting Data to train sets and Applying BOX COX Transformation to introduce<br>Linearity among Variables . . . . . | 4         |
| <b>4</b> | <b>Research</b>  | <b>7</b>  |
| 4.1      | Training Transformed Data and Performing Linear regression . . . . .   | 7         |
| 4.2      | Prediction and Cross Validation over Transformed Data . . . . .  | 8         |
| <b>5</b> | <b>Research</b>  | <b>9</b>  |
| 5.1      | Muticollinearity and VIF . . . . .   | 9         |
| 5.2      | Variance Inflation factor on Transformed Data . . . . .  | 9         |
| 5.2.1    | Performing PCA to reduce Dimensions and variables . . . . .  | 9         |
| <b>6</b> | <b>Research</b>  | <b>11</b> |
| 6.1      | Partial Linear Square . . . . .  | 11        |
| 6.2      | Results of PLS . . . . .   | 11        |
| <b>7</b> | <b>Conclusions</b>   | <b>13</b> |

# Chapter 1

## Introduction

### 1.1 OVERFITTING

Overfitting is a modeling error that occurs when a function is too closely fit to a limited set of data points. Overfitting the model generally takes the form of making an overly complex model to explain idiosyncrasies in the data under study. As I experienced with Decision Trees on the data

### 1.2 BOX-COX

This is a useful data transformation technique used to stabilize variance, make the data more normal distribution-like, improve the validity of measures of association such as the Pearson correlation between variables and for other data stabilization procedures.

At the core of the Box Cox transformation is an exponent,  $\lambda$ , which varies from -5 to 5. All values of  $\lambda$  are considered and the optimal value for your data is selected; The “optimal value” is the one which results in the best approximation of a normal distribution curve. The transformation of  $Y$  has the form:

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0; \\ \log y, & \text{if } \lambda = 0. \end{cases}$$

Figure 1.1: BOX COX FORMULA

### 1.3 PRINCIPAL COMPONENT ANALYSIS

In the Call Housing Data, there are independent variables that are co-dependent on each other and when running their correlation matrix, might see that there is a high correlation between each other. When such both variables are included in the regression model this will be like the fact that much of the variance of one of the variables is already been captured by the other variable.

Lets assume that two variables are correlated as 0.69 hence when we include both of these variables in the regression model, then the 69% of the variance is already accounted by one of the variables for the model, hence adding the other feature will not add any additional value. This is especially useful in Call Housing case when we have a huge count of independent variables and we need to reduce the count of the model independent variables, and make our model more compact with limited set of the independent variables. As there are as many principal components as there are variables in the data, principal components are constructed in such a manner that the first principal component accounts for the largest possible variance in the data set. The second principal component is calculated in the same way, with the condition that it is uncorrelated with (i.e., perpendicular to) the first principal component and that it accounts for the next highest variance.

This continues until a total of  $p$  principal components have been calculated, equal to the original number of variables.

## 1.4 Partial Least Square

Partial least squares regression (PLS regression) is a statistical method that bears some relation to principal components regression; instead of finding hyperplanes of maximum variance between the response and independent variables, it finds a linear regression model by projecting the predicted variables and the observable variables to a new space.

PLS is used to find the fundamental relations between two matrices (X and Y).

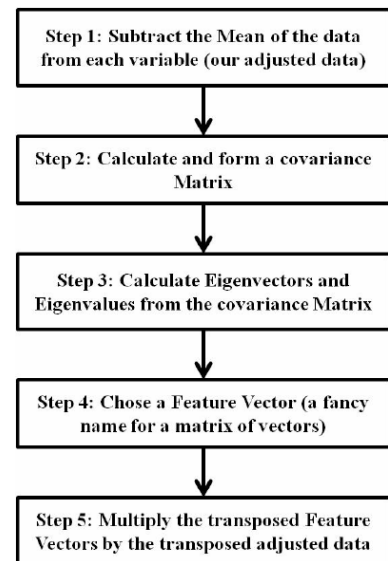


Figure 1.2: PCA ALGORITHM

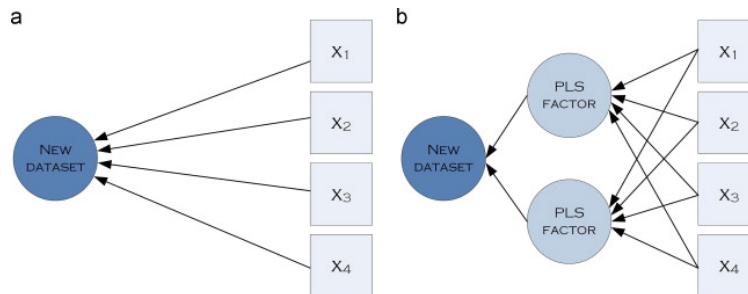


Figure 1.3: PLS ON VARIABLES VS SIMPLE LINEAR REGRESSION

## 1.5 Variance Inflation Factor

A variance inflation factor(VIF) detects multicollinearity in regression analysis. Multicollinearity is when there's correlation between predictors (i.e. independent variables) in a model; it's presence can adversely affect your regression results. The VIF estimates how much the variance of a regression coefficient is inflated due to multicollinearity in the model.

## Chapter 2

# REDUCING MULTI-COLLINEARITY ,OVERFITTING AND LINEARIZING with R

### 2.1 Call Housing Data

As from the previous report of Decision tree ,I was able to predict the Houses values with good SSE through Decision Tress,but according to me those prediction couldnt be reliable because the results shows Overfitting, and the variables had too much noise and variance plus most of the variable were corelating to each other ,which leads to major multicollinearity.

So I will try to use some Unsupervised Statistics Methods over the data in order to refine it.

What I tried to Do in this Project-

- Firstly I Manipulate the data and checked for any skewness.
- Then I performed a box-cox transformation,in order to make variables more Linear
- Then I performed Ordinary least squares regression over the transformed data. Then I used ridge estimates to check for multicollinearity.
- Then Using principal component analysis I performed regression using optimal components to reduce residual error.
- Than to further improve the model I perform Partial least square technique in order to further reduce crrelation and co dependence so data may reduce OVERFITTING

# Chapter 3

## Research

### 3.1 Scatter-plot matrix to see Linearity among Variables

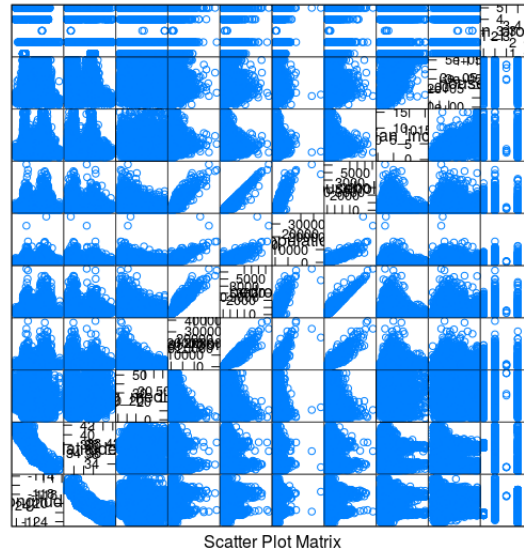


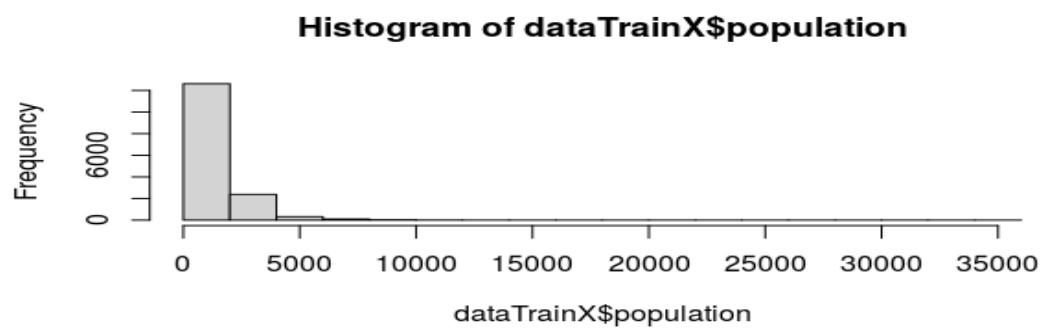
Figure 3.1: Appendix Reference-1

Matrix names Sequence -longitude,latitude ,housing\_median\_age, total\_rooms, total\_bedrooms, population ,households, median\_income, median\_house\_value, ocean\_proximity

As from Scatter plot Matrix ,we can see many independent Variables are Linear and correlated to each other,like Total rooms and Total Bedrooms

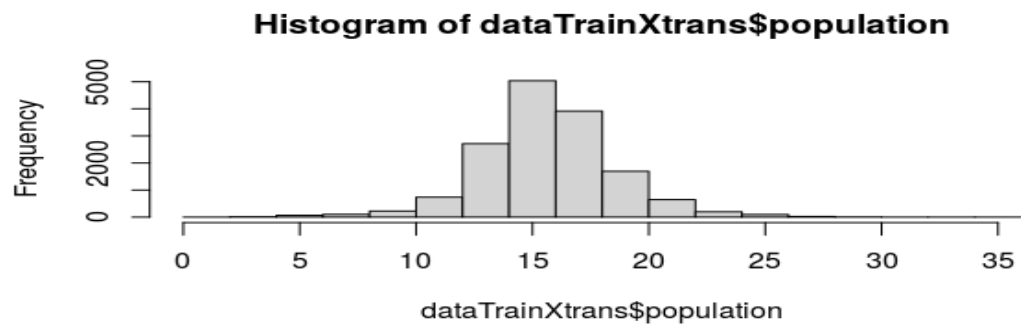
For our Houses values ,we see linearity only with median\_income and with rest of variables there is lot of variance and need to be fixed.

#### 3.1.1 Splitting Data to train sets and Applying BOX COX Transformation to introduce Linearity among Variables



0.20pt

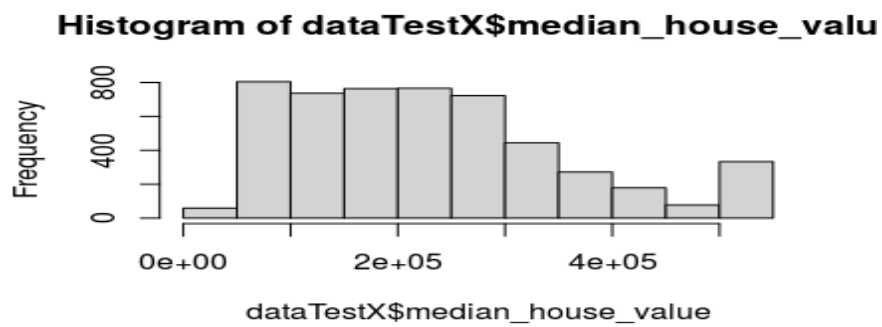
Figure 3.2: Before BOX COX



0.20pt

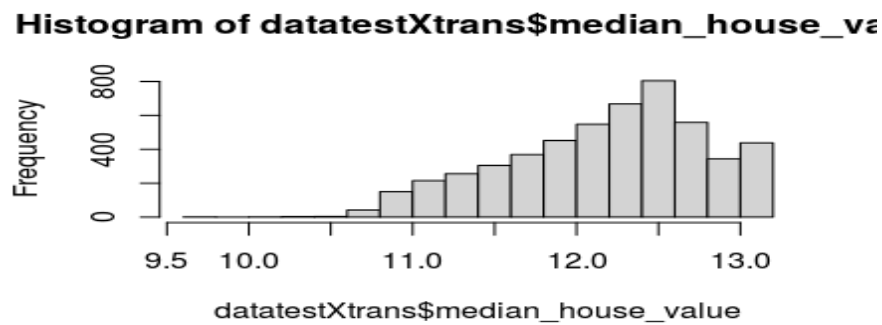
Figure 3.3: After BOX COX

Figure 3.4: Comparison before and after BOXCOX of Population Variable



0.20pt

Figure 3.5: Before BOX COX



0.20pt

Figure 3.6: After BOX COX

Figure 3.7: Comparison before and after BOXCOX of medaian\_house\_value Variable



# Chapter 4

## Research

### 4.1 Training Transformed Data and Performing Linear regression

```
lm(formula = median_house_value ~ ., data = trainingData)

Residuals:
    Min       1Q   Median       3Q      Max
-399104  -41817  -11719   26951   784835

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.730e+06  8.198e+04 -33.296 < 2e-16 ***
longitude    -3.177e+04  9.426e+02 -33.705 < 2e-16 ***
latitude     -3.056e+04  9.235e+02 -33.096 < 2e-16 ***
housing_median_age  9.211e+02  5.038e+01  18.282 < 2e-16 ***
total_rooms   -6.540e+00  8.682e-01  -7.533 5.22e-14 ***
total_bedrooms  9.861e+01  7.362e+00  13.395 < 2e-16 ***
population    -3.786e+01  1.180e+00 -32.091 < 2e-16 ***
households     5.060e+01  8.021e+00   6.308 2.91e-10 ***
median_income  3.992e+04  3.842e+02 103.909 < 2e-16 ***
ocean_proximity 1.028e+04  4.780e+02  21.516 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 67530 on 15320 degrees of freedom
(150 observations deleted due to missingness)
Multiple R-squared:  0.6327,    Adjusted R-squared:  0.6325
F-statistic: 2933 on 9 and 15320 DF,  p-value: < 2.2e-16
```

Figure 4.1: Linear Regression on Power Transformed Trained Data

As compared to Linear regression performed in previous report and after transforming data to make it more Linear, results have been improved with increase in R squared from 0.24 to 0.63

```

Call:
lm(formula = median_house_value ~ latitude + longitude, data = dat

Residuals:
    Min       1Q   Median       3Q      Max
-316022  -67502  -22903   46042  483381

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -5829397.0     82092.2   -71.01  <2e-16 ***
latitude     -69551.0       859.6    -80.91  <2e-16 ***
longitude     -71209.4       916.4    -77.70  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 100400 on 20637 degrees of freedom
Multiple R-squared:  0.2424,    Adjusted R-squared:  0.2423
F-statistic: 3302 on 2 and 20637 DF,  p-value: < 2.2e-16

```

Figure 4.2: Linear Regression Performed on the data in Decision tree report

## 4.2 Prediction and Cross Validation over Transformed Data

```

> detailSummary(UT)
              RMSE      Rsquared      MAE
1.223050e+06 2.883152e-01 1.221919e+06

```

Figure 4.3: Prediction on Training Data Results

```

Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 13931, 13932, 13932, 13932,
Resampling results:

      RMSE      Rsquared      MAE
0.3206073 0.6645923 0.2403438

```

Figure 4.4: Cross Validation Results

## Chapter 5

# Research

### 5.1 Muticollinearity and VIF

Further I will try to improve the model by checking for mutlicollinearity and if found, Performing PCA and PLS to remove collinearity

### 5.2 Variance Inflation factor on Transformed Data

```
vif(model)
      longitude      latitude housing_median_age      total_rooms      total_bedrooms
10.108791      12.032526      1.314938      13.354221      35.512144
population
6.529896      households      median_income      ocean_proximity
34.358679      1.766226      1.859728
```

Figure 5.1: Vif results

The levels of VIF shows Collinearity among Variables

#### 5.2.1 Performing PCA to reduce Dimensions and variables

```
Importance of components:
      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8      PC9
Standard deviation  1.9798 1.5117 1.0421 0.92476 0.76473 0.38187 0.26521 0.20976 0.12176
Proportion of Variance 0.4352 0.2538 0.1206 0.09496 0.06494 0.01619 0.00781 0.00489 0.00165
Cumulative Proportion 0.4352 0.6890 0.8096 0.90453 0.96947 0.98566 0.99347 0.99835 1.00000
```

Figure 5.2: PCA COMPONENTS IMPORTANCE

Running the principal component analysis shows that after adding the 5th variables this has already accounted for 97% (0.96947) of the variance that were expecting, and I can run the revised model with only five parameters and I would get significant results as well for our multiple linear regression model. The option `scale.=T`(Appendix) lets me make the data normalized, which is important where some features are off scale.

|       | obs    | median_house_value.5.comps |
|-------|--------|----------------------------|
| 15481 | 193200 | 172698.2                   |
| 15482 | 168400 | 175040.0                   |
| 15483 | 174600 | 173715.9                   |
| 15484 | 158900 | 172754.6                   |
| 15485 | 172600 | 173809.3                   |
| 15486 | 170700 | 173796.3                   |
| 15487 | 171400 | 160789.0                   |
| 15488 | 163600 | 173784.0                   |
| 15489 | 172600 | 169072.1                   |
| 15490 | 153900 | 169055.0                   |
| 15491 | 92000  | 168854.5                   |
| 15492 | 117500 | 162886.0                   |
| 15493 | 168000 | 177396.0                   |
| 15494 | 247800 | 176151.6                   |
| 15495 | 258900 | 159577.6                   |

Figure 5.3: Real Data VS Predictions After PCA

The Model have been improved significantly, but still it is overestimating for the cheaper houses. Let's Try if Performing Partial Linear Square, which is extension to PCA concept, and can be better in Selecting weights, which may remove the shortcomings of the model from PCA (overestimating for cheaper houses)

## Chapter 6

# Research

### 6.1 Partial Linear Square

According to partial Linear Square,the perfect components will be 4 for the predictions

```
Data:  X dimension: 15330 9
      Y dimension: 15330 1
Fit method: kernelpls
Number of components considered: 9

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
CV          111395 110249 106063 105051 103464 100508 80936 69270 67693 67678
adjCV       111395 110249 106058 105075 103454 100502 80920 69262 67679 67669

TRAINING: % variance explained
      1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps 8 comps 9 comps
X          93.608 99.499 99.88 100.00 100.00 100.00 100.0 100.00 100.00
median_house_value 2.065 9.509 11.13 14.03 18.86 47.51 61.5 63.27 63.27
> |
```

Figure 6.1: PLS COMPONENTS

### 6.2 Results of PLS

Though the results have been improved from the PCA predictions,with reduced RMSE, and incresed R squared.But still there are some problems and shortcomings of UNSUPERVISED MODELS,that I will discuss in **Conclusions**

```
>
>
> defaultSummary(pls.eval)
      RMSE      Rsquared      MAE
1.216593e+05 6.407495e-02 9.628502e+04
> |
```

Figure 6.2: PLS Results - RMSE IS REDUCED SIGNIFICANTLY




|  | obs  | pred  |
|---|---|--|
| 15481   | 193200  | 184077.4   |
| 15482   | 168400  | 189135.0   |
| 15483   | 174600  | 186667.9   |
| 15484   | 158900  | 183461.7   |
| 15485   | 172600  | 186044.3   |
| 15486   | 170700  | 186692.3   |
| 15487   | 171400  | 162519.2   |
| 15488   | 163600  | 186441.9   |
| 15489   | 172600  | 177662.3   |
| 15490   | 153900  | 177704.3   |
| 15491   | 92000   | 176621.1   |
| 15492   | 117500  | 165691.2   |
| 15493   | 168000  | 191920.8   |
| 15494   | 247800  | 190177.7   |
| 15495   | 258900  | 159466.8   |

Figure 6.3: Real data vs PLS predictions

## Chapter 7

# Conclusions

Though I was able to improve the model significantly and able to remove Variance, multicollinearity from the data significantly, which lead to improvement of Results but still

- Model could not evaluate expensive Areas
- Model could not evaluate correctly very cheap areas

**According to me these problems in Model could be**

- In data there is a data entry mistake, I studied the data and there is a value 500001 of median.housechic is randomly assigned to multiple areas, with variables similar to cheap or expensive areas ,
- I may need to add other variables ,that do not depend on Supervised Learning or unsupervised Learning but more to the empirical studies, For example like maybe in some areas which are on "ISLAND" need to get some bias because ,on that ISLAND only exclusive rich or celebrities live and that is why the house values is too expensive because of that bias,
- or maybe some variable like CRIME RATE is missing and I need to a bias or research on it to calculate why there is difference of prices of houses of the areas having similar Variable Values
- Maybe I can try to use Tensorflow Deep learning to predict the areas with different prices of house but similar other variables, which can be better option than K NEAREST NEIGHBOURS

# Appendix

## 1

```
library(caret)
library(AppliedPredictiveModeling)
library(pls)
library(e1071)
library(lattice)
library(pls)
library(MASS)
library(lars)
library(elasticnet)
library(car)
library(glmnet)
library(plyr)

###read data
data<-read.csv("cal-housing_1.csv")
head(data)
data$ocean_proximity<-revalue(data$ocean_proximity, c("NEAR_BAY"="1", "INLAND"="2", "ISLAND"="3", "NEAR_OCEAN"="4", "<1H_OCEAN"="5"))
data$ocean_proximity <- as.numeric(data$ocean_proximity)
data<-data[,!names(data) %in% c('X','X.1')]

###do a scatterplot matrix to see linearilty
splom(data)
nrow(data)
head(data)
```



```

###create training and test data set
###set 75 percent of rows for training and rest for test
bound<-floor(0.75*nrow(data))
data.train <- data[1:bound, ]
data.test <- data[(bound+1):nrow(data), ]
nrow(data.test)
nrow(data.train)
dataTrainX<-data.train
dataTestX<-data.test

###apply box cox transformation
boxcox<-preProcess(dataTrainX,method ="BoxCox")
dataTrainXtrans<-predict(boxcox,dataTrainX)
head(dataTrainXtrans)
hist(dataTrainXtrans$population)
hist(dataTrainX$population)

datatestXtrans<-predict(boxcox,dataTestX)
head(datatestXtrans)
hist(datatestXtrans$median_house_value)
hist(dataTestX$median_house_value)

###create training data
trainingData<-dataTrainXtrans
trainingData<-dataTrainX
head(trainingData)

```

```

###fit the model-OLS
model<-lm(median_house_value~.,data=trainingData)
summary(model)
par(mfrow=c(2,2))

###predict values
pred<-predict(model,datatestXtrans)
###create obs,pred data frame
df<-data.frame(obs=datatestXtrans$median_house_value,pred=pred)
df
defaultSummary(df)
###cross-validation
ctrl<-trainControl(method="cv",n=10)
set.seed(100)
tmp<-subset(dataTrainXtrans,select =-median_house_value)
head(tmp)
modcv<-train(x=tmp,y=dataTrainXtrans$median_house_value,method="lm",trControl =ctrl)

###check for multicollinearity
vif(model)
###vif levels shows collinearity in the dataset

###pca analysis
pca<-data
###standardize independent variables
x<-subset(pca,select=-median_house_value)

```

```

head(x)
x<-scale(x)
###center the dependent variable
y<-pca$median_house_value
y<-scale(y,scale =F)
###do pca on indepenedent variables
comp<-prcomp(na.omit(x))
comp
plot(comp)
biplot(comp)
summary(comp)
#5 principal components explain 97% of the total variance

pcr<-pcr(median_house_value~.,data=trainingData,validation="CV")
summary(pcr)
###choose five components for prediction
xpcr=subset(datatestXtrans,select=-median_house_value)
pcrpred<-predict(pcr,xpcr,ncomp =5)
pcrdf1<-data.frame(obs=dataTestX$median_house_value,Predictions=pcrpred)
pcrdf1

###pls regression is a better variation of PCR.It accounts for the variation in response when selecting weights
###use pls package, pls function
###default algorithm is Dayal and Mcgregor kernel algorithm
plsFit<-plsr(median_house_value~.,data=trainingData,validation="CV")
###predict first five median_house_value values using 1 and 2 components
pls.pred<-predict(plsFit,datatestXtrans[1:100,],ncomp=1:2)
summary(plsFit)

```

```

validationplot(plsFit, val.type = "RMSEP")
pls.RMSEP <- RMSEP(plsFit, estimate = "CV")
plot(pls.RMSEP, main = "RMSEP vs PLS", xlab = "Components")
min <- which.min(pls.RMSEP$val)
points(min, min(pls.RMSEP$val), pch = 1, col = "red")
plot(plsFit, ncomp = 4, asp = 1)

### use 4 components
pls.pred2 <- predict(plsFit, datatestXtrans, ncomp = 5)
pls.eval <- data.frame(obs = dataTestX$median_house_value, pred = pls.pred2[, 1, 1])
defaultSummary(pls.eval)

```

# List of Figures

|     |  |    |
|-----|--|----|
| 1.1 | BOX COX FORMULA . . . . .  | 1  |
| 1.2 | PCA ALGORITHM . . . . .  | 2  |
| 1.3 | PLS ON VARIABLES VS SIMPLE LINEAR REGRESSION . . . . .                       | 2  |
| 3.1 | Appendix Reference-1 . . . . .   | 4  |
| 3.2 | Before BOX COX . . . . .   | 5  |
| 3.3 | After BOX COX . . . . .  | 5  |
| 3.4 | Comparison before and after BOXCOX of Population Variable . . . . .          | 5  |
| 3.5 | Before BOX COX . . . . .   | 6  |
| 3.6 | After BOX COX . . . . .  | 6  |
| 3.7 | Comparison before and after BOXCOX of medaian_house_value Variable . . . . . | 6  |
| 4.1 | Linear Regression on Power Transformed Trained Data . . . . .                | 7  |
| 4.2 | Linear Regression Performed on the data in Decision tree report . . . . .    | 8  |
| 4.3 | Prediciton on Training Data Results . . . . .                                | 8  |
| 4.4 | Cross Validation Results . . . . .   | 8  |
| 5.1 | Vif results . . . . .  | 9  |
| 5.2 | PCA COMPONENTS IMPORTANCE . . . . .  | 9  |
| 5.3 | Real Data VS Predictions After PCA . . . . .                                 | 10 |
| 6.1 | PLS COMPONENTS . . . . .   | 11 |
| 6.2 | PLS Results - RMSE IS REDUCED SIGNIFICANTLY . . . . .                        | 11 |
| 6.3 | Real data VS PLS predictions . . . . .                                       | 12 |