USE CASE STUDY REPORT

Group No.: Group 2 (Advanced case study)

Student Names: Aditya Pradeep Sohoni, Jatin Ajmera, Neelakantha Dolai, Nishka Ranjan and Shanay Shah.

Executive Summary

Background and Introduction: -

Analysis of heating load and cooling load is paramount to check the energy consumption of a building. We attempt to determine and establish relationship between the heating and cooling load and the building parameters. We use 12 different building shapes simulation in Ecotect (a feature of Autodesk). Most importantly, the buildings differ with respect to the glazing area distribution, and the orientation, amongst other parameters. We will simulate various settings as functions of the aforementioned characteristics to obtain 768 building shapes.

Origin of Data:-

The dataset comprises 768 samples and 8 features, aiming to predict two real valued responses. It can also be used as a multi-class classification problem if the response is rounded to the nearest integer. The data set was created in November 2012 and to proceed with this we would prefer to use Machine Learning methods to perform the analysis. Our main aim was to select an industrial machine analysis data.

Problems: -

- Assessing heating load and cooling load requirements of buildings (i.e. energy efficiency) as a function of building parameters.
- Studying the effect of eight input variables (relative compactness, surface area, wall area, roof area, overall height, orientation, glazing area, glazing area distribution) on two output variables, namely heating load (HL) and cooling load (CL), of residential buildings.
- Identifying the most strongly related input variables.
- Comparing a classical linear regression approach against a powerful state of the art nonlinear nonparametric method.
- Predict HL and CL with low mean absolute error deviations from the ground truth.
- Results of this study support the feasibility of using machine learning tools to estimate building parameters as a convenient and accurate approach.

Goals of the study:-

- For efficiency, heating load (HL) and the cooling load (CL) is required to determine the specifications of the heating and cooling equipment.
- Develop a model to conserve energy .
- Develop an efficient energy system

- Concerns about energy waste and its perennial adverse impact on the environment.
- Building energy consumption has steadily increased over the past decades worldwide. Heating, ventilation and air conditioning account for maximum energy usage.
- Desirable to have more energy-efficient building designs with improved energy conservation properties.
- Using advanced dedicated building energy simulation software can be very time-consuming, requires user-expertise in a particular program, accuracy may vary.

Solution -

- Using statistical and machine learning is an extremely fast process to obtain answers by varying some building design parameters once a model has been adequately trained.

II. Data Exploration and Visualization

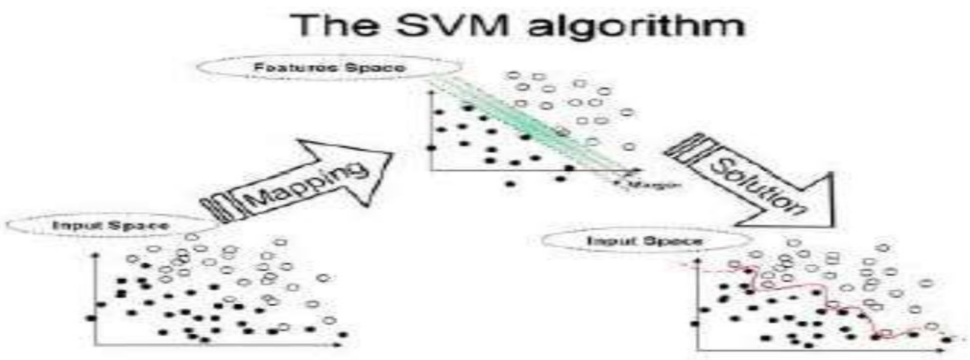
These are the techniques used:

1) **Cluster Analysis-** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is our main task as it helped to segregate parameter 1,2 and 3 in one group as they were more close to each other than the rest 5 parameters.

2) **Multiple Regression**- It was quite suitable to use in our scenario as it emphasize the need to split the data into two categories: the training data set and the validation data set to be able to validate the multiple linear regression model, and the need to relax the assumption that errors follow a Normal distribution. After this review, we introduced methods for identifying subsets of the independent variables to improve predictions. We generated the following results for the hot load condition by splitting the independent variables into two parts of 5 and 3.

3) **Support Vector Machine Regression**

Kernel methods in general have gained increased attention in recent years, partly due to the grown of popularity of the Support Vector Machines. Support Vector Machines are linear classifiers and repressors that, through the Kernel trick, operate in reproducing Kernel Hilbert spaces and are thus able to perform nonlinear classification and regression in their input space.



4) **Principal Component Analysis**
It is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called **principal components.**

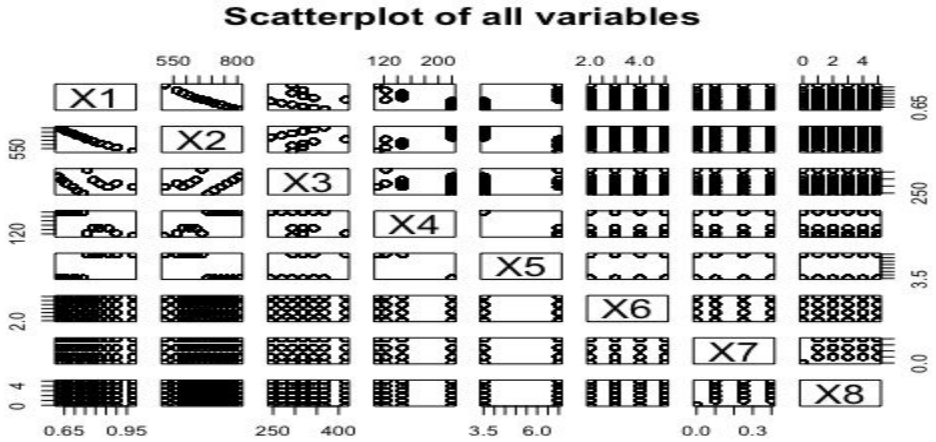
5) **Least Square methods**
The most important application is in data fitting. The best fit in the least-squares sense minimizes the sum of squared residuals (a residual being: the difference between an observed value, and the fitted value provided by a model).

6) Clustering using SimpleKMeans
It is simple clustering algorithm. It partitions n data tuples into k groups such that each entity in the cluster has nearest mean. This paper is about the implementation of the clustering techniques using WEKA interface.

III. Data Preparation and Preprocessing

Provide information on data summary, dimension reduction, correlation analysis, PCA analysis, variable converting, variable selection, etc.
Data Description:
The dataset contains eight attributes (or features, denoted by X1...X8) and two responses (or outcomes, denoted by y1 and y2). The aim is to use the eight features to predict each of the two responses. The dataset includes 768 samples and 8 features, aiming to predict two real valued responses. It can also be used as a multi-class classification problem if the response is rounded to the nearest integer. The data set was created in November 2012.
Specifically:
X1 Relative Compactness
X2 Surface Area
X3 Wall Area
X4 Roof Area
X5 Overall Height
X6 Orientation
X7 Glazing Area

X8 Glazing Area Distribution
y1 Heating Load
y2 Cooling Load

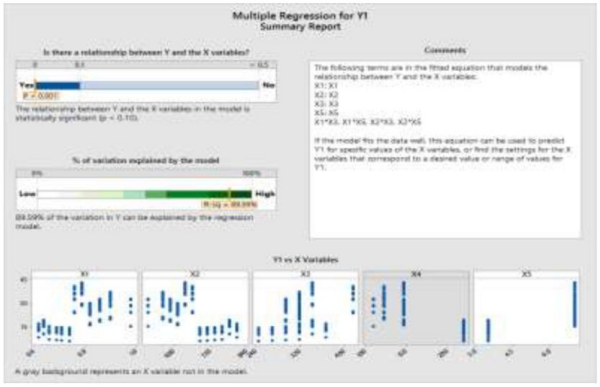


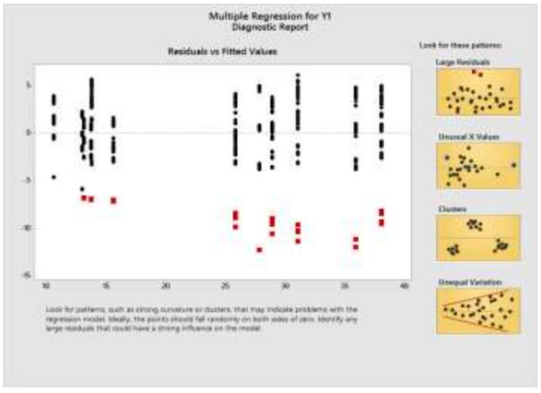
IV. Data Mining Techniques and Implementation

Multiple Regression- It was quite suitable to use in our scenario as it emphasize the need to split the data into two categories: the training data set and the validation data set to be able to validate the multiple linear regression model, and the need to relax the assumption that errors follow a Normal distribution

A).The Summary Report tells us that our regression model is statistically significant with a P value less than 0.001 and has an R squared value of 89.59%.

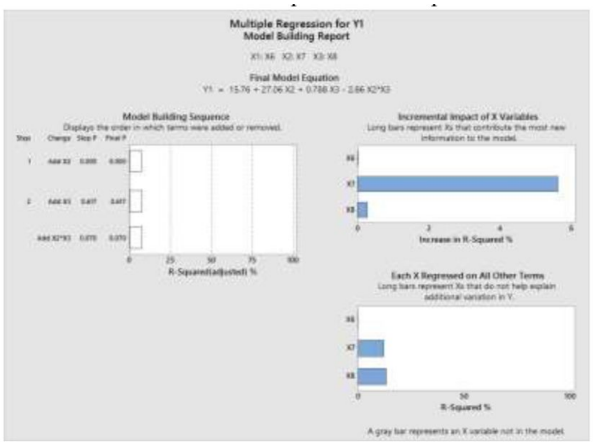
b) The Diagnostic Report displays the residuals versus fitted values and identifies unusual points that we should investigate.

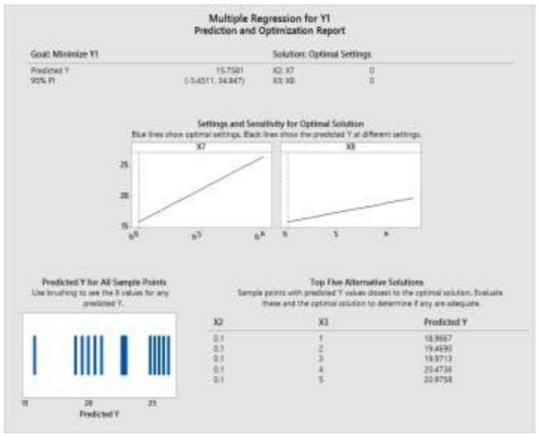




C)The Model Building Report shows the details about how the regression model is built, the regression equation, which variables contribute the most information, and whether the X variables are correlated with each other.

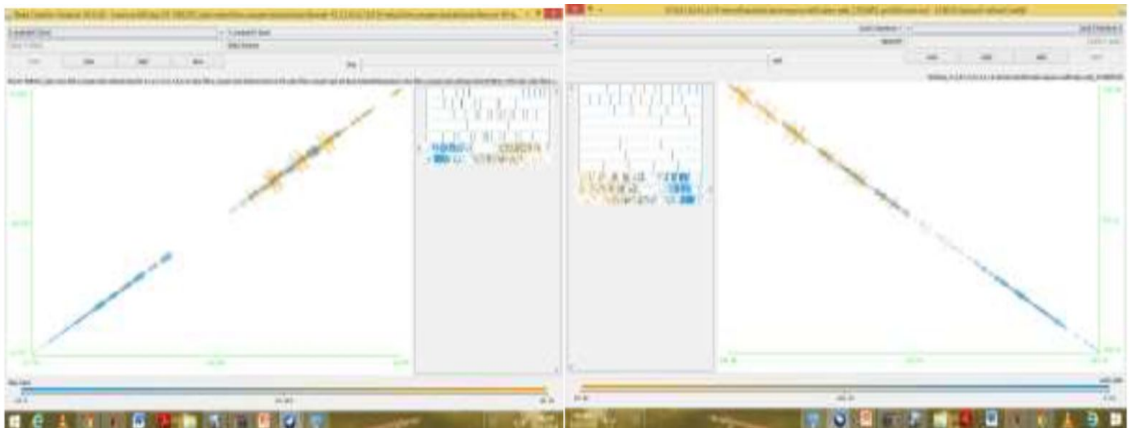
D) Prediction and Optimisation Report shows that the x variables define the predicted y values.



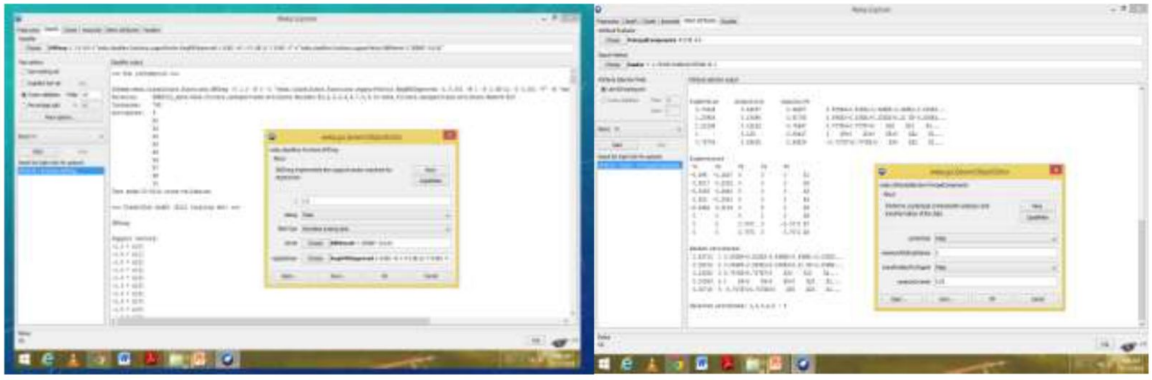


Analysis using WEKA:-

Weka supports several standard data mining tasks, more specifically, data preprocessing, clustering, classification, regression, visualization, and feature selection . All of Weka's techniques are predicated on the assumption that the data is available as one flat file or relation, where each data point is described by a fixed number of attributes (normally, numeric or nominal attributes, but some other attribute types are also supported).



Now in the below diagrams, we can observe that how the points have been scattered around but in the uppermost model it has streamlined by using an optimised model run through WEKA.



Analysis Using PCA:-

1. Summarization of data with many (p) variables by a smaller set of (k) derived (synthetic, composite) variables.

2. It takes a data matrix of n objects by p variables, which may be correlated, and summarizes it by uncorrelated axes (principal components or principal axes) that are linear combinations of the original p variables

3. The first k components display as much as possible of the variation among objects.

V. Performance Evaluation

Multiple regression methods is best to find relationship between variables. we found out strongly related variables using multiple regression method. Then, we compared a classical linear regression approach against a powerful state of the art nonlinear nonparametric method, random forests, to estimate HL and CL.

VI. Discussion and Recommendation

So we reached following conclusions after our project.
• Importance of heating and cooling system
• Always a challenge to develop a model to conserve the energy.
• Living in extreme climate we are aware of the heating and cooling system.
• Apart from heating and cooling, developing an efficient energy system is most important concern
of the case study.

Future Scope of the Project:
• The methodology can be extended to encompass additional input variables namely:
1) Environmental Climate
2) Occupancy of the Residential Building.
• Similarly, additional output variables could be studied using the approach developed.
• To estimate the required cooling and heating capacities, architects and building designers need information about the parameters of the building.

VII. Summary

We developed a statistical machine learning framework to study the effect of eight input variables (relative compactness, surface area, wall area, roof area, overall height, orientation, glazing area, glazing area distribution) on two output variables, namely heating load (HL) and cooling load (CL), of residential buildings. We systematically investigated the association strength of each input variable with each of the output variables using a variety of classical and non-parametric statistical analysis tools, in order to identify the most strongly related input variables. Then, we compared a classical linear regression approach against a powerful state of the art nonlinear nonparametric method, random forests, to estimate HL and CL. The results of this study supported the feasibility of using machine learning tools to estimate building parameters as a convenient and accurate approach, as long as the requested query bears resemblance to the data actually used to train the mathematical model in the first place.

Appendix: R Code for use case study

Please show the R code you generated for the use case study. Please do not show results here, only the code.

```
> ## decision tree
> library(caret)
Loading required package: lattice
Loading required package: ggplot2
> library(klaR)
Loading required package: MASS
> # load the  dataset
> energydata <- read.csv("/Users/nishkaranjan/Downloads/energydata.csv")
> # define an 80%/20% train/test split of the dataset
> split=0.80
> trainIndex <- createDataPartition(energydata$Y1, p=split, list=FALSE)
> data_train <- energydata[ trainIndex,]
> data_test <- energydata[-trainIndex,]
>
> library(party)
Loading required package: grid
Loading required package: mvtnorm
Loading required package: modeltools
Loading required package: stats4
Loading required package: strucchange
Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

        as.Date, as.Date.numeric

Loading required package: sandwich
> output.forest <- ctree(Y1 ~ X1 + X2+ X3 + X4+ X5+ X6 + X7 + X8,
+                        data = energydata)
> plot(output.forest, uniform=T)
> # make predictions
> x_test <- data_test[,1:8]
> y_test <- data_test[,9]
> predictions <- predict(output.forest, x_test)
>
>
> install.packages("e1071")
trying URL 'https://cran.rstudio.com/bin/macosx/mavericks/contrib/3.3/e1071_1.6-8.tgz'
Content type 'application/x-gzip' length 742779 bytes (725 KB)
```

```
==================================================
downloaded 725 KB

tar: Failed to set default locale

The downloaded binary packages are in
/var/folders/fv/67bd79_n4_14lpl74rl0yhj00000gn/T//RtmpEnCMO9/downloaded_packag
es
>
> # summarize results
> #Use this for class(factors like 1 , 2) variables
> confusionMatrix(factor(round(predictions)), y_test)
Confusion Matrix and Statistics

          Reference
Prediction  0  1  2
         0 46  3  0
         1  5 59  0
         2  0  0 40

Overall Statistics

               Accuracy : 0.9477
                 95% CI : (0.8996, 0.9772)
    No Information Rate : 0.4052
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9202
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.9020   0.9516   1.0000
Specificity            0.9706   0.9451   1.0000
Pos Pred Value         0.9388   0.9219   1.0000
Neg Pred Value         0.9519   0.9663   1.0000
Prevalence             0.3333   0.4052   0.2614
Detection Rate         0.3007   0.3856   0.2614
Detection Prevalence   0.3203   0.4183   0.2614
Balanced Accuracy      0.9363   0.9483   1.0000
> ## decision tree
> library(caret)
> library(klaR)
> # load the  dataset
```

```
> energydata <- read.csv("/Users/nishkaranjan/Downloads/energydata.csv")
> # define an 80%/20% train/test split of the dataset
> split=0.80
> trainIndex <- createDataPartition(energydata$Y1, p=split, list=FALSE)
> data_train <- energydata[ trainIndex,]
> data_test <- energydata[-trainIndex,]
>
> library(party)
> output.forest <- ctree(Y2 ~ X1 + X2+ X3 + X4+ X5+ X6 + X7 + X8,
+                        data = energydata)
> plot(output.forest, uniform=T)
> # make predictions
> x_test <- data_test[,1:8]
> y_test <- data_test[,9]
> predictions <- predict(output.forest, x_test)
>
>
> install.packages("e1071")
Error in install.packages : Updating loaded packages
>
> # summarize results
> #Use this for class(factors like 1 , 2) variables
> confusionMatrix(factor(round(predictions)), y_test)
Confusion Matrix and Statistics
 Reference
Prediction  0  1  2
          0 25  0  0
          1 28 44  0
          2  0 16 40
Overall Statistics
                                          Accuracy : 0.7124
                 95% CI : (0.6338, 0.7826)
    No Information Rate : 0.3922
    P-Value [Acc > NIR] : 1.048e-15

                  Kappa : 0.5663
 Mcnemar's Test P-Value : NA

Statistics by Class:
 Class: 0 Class: 1 Class: 2
Sensitivity          0.4717   0.7333   1.0000
Specificity          1.0000   0.6989   0.8584
Pos Pred Value       1.0000   0.6111   0.7143
Neg Pred Value       0.7812   0.8025   1.0000
Prevalence           0.3464   0.3922   0.2614
```

```
Detection Rate         0.1634   0.2876   0.2614
Detection Prevalence   0.1634   0.4706   0.3660
Balanced Accuracy      0.7358   0.7161   0.9292
> install.packages("e1071")

2nd
> ## decision tree
> library(caret)
Loading required package: lattice
Loading required package: ggplot2
> library(klaR)
Loading required package: MASS
> # load the  dataset
> energydata <- read.csv("/Users/nishkaranjan/Downloads/energydata.csv")
> # define an 80%/20% train/test split of the dataset
> split=0.80
> trainIndex <- createDataPartition(energydata$Y1, p=split, list=FALSE)
> data_train <- energydata[ trainIndex,]
> data_test <- energydata[-trainIndex,]
>
> library(party)
Loading required package: grid
Loading required package: mvtnorm
Loading required package: modeltools
Loading required package: stats4
Loading required package: strucchange
Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

        as.Date, as.Date.numeric

Loading required package: sandwich
> output.forest <- ctree(Y1 ~ X1 + X2+ X3 + X4+ X5+ X6 + X7 + X8,
+                        data = energydata)
> plot(output.forest, uniform=T)
> # make predictions
> x_test <- data_test[,1:8]
> y_test <- data_test[,9]
> predictions <- predict(output.forest, x_test)
>
>
> install.packages("e1071")
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/mavericks/contrib/3.3/e1071_1.6-8.tgz'
Content type 'application/x-gzip' length 742779 bytes (725 KB)
==================================================
downloaded 725 KB

tar: Failed to set default locale

The downloaded binary packages are in
/var/folders/fv/67bd79_n4_14lpl74rl0yhj00000gn/T//RtmpEnCMO9/downloaded_packag
es
>
> # summarize results
> #Use this for class(factors like 1 , 2) variables
> confusionMatrix(factor(round(predictions)), y_test)
Confusion Matrix and Statistics

          Reference
Prediction  0  1  2
         0 46  3  0
         1  5 59  0
         2  0  0 40

Overall Statistics

               Accuracy : 0.9477
                 95% CI : (0.8996, 0.9772)
    No Information Rate : 0.4052
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.9202
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity            0.9020   0.9516   1.0000
Specificity            0.9706   0.9451   1.0000
Pos Pred Value         0.9388   0.9219   1.0000
Neg Pred Value         0.9519   0.9663   1.0000
Prevalence             0.3333   0.4052   0.2614
Detection Rate         0.3007   0.3856   0.2614
Detection Prevalence   0.3203   0.4183   0.2614
Balanced Accuracy      0.9363   0.9483   1.0000
> ## decision tree
> library(caret)
```

```
> library(klaR)
> # load the  dataset
> energydata <- read.csv("/Users/nishkaranjan/Downloads/energydata.csv")
> # define an 80%/20% train/test split of the dataset
> split=0.80
> trainIndex <- createDataPartition(energydata$Y1, p=split, list=FALSE)
> data_train <- energydata[ trainIndex,]
> data_test <- energydata[-trainIndex,]
>
> library(party)
> output.forest <- ctree(Y2 ~ X1 + X2+ X3 + X4+ X5+ X6 + X7 + X8,
+                        data = energydata)
> plot(output.forest, uniform=T)
> # make predictions
> x_test <- data_test[,1:8]
> y_test <- data_test[,9]
> predictions <- predict(output.forest, x_test)
>
>
> install.packages("e1071")
Error in install.packages : Updating loaded packages
>
> # summarize results
> #Use this for class(factors like 1 , 2) variables
> confusionMatrix(factor(round(predictions)), y_test)
Confusion Matrix and Statistics

          Reference
Prediction  0  1  2
         0 25  0  0
         1 28 44  0
         2  0 16 40

Overall Statistics

               Accuracy : 0.7124
                 95% CI : (0.6338, 0.7826)
    No Information Rate : 0.3922
    P-Value [Acc > NIR] : 1.048e-15

                  Kappa : 0.5663
 Mcnemar's Test P-Value : NA

Statistics by Class:
```

```
                     Class: 0 Class: 1 Class: 2
Sensitivity            0.4717   0.7333   1.0000
Specificity            1.0000   0.6989   0.8584
Pos Pred Value         1.0000   0.6111   0.7143
Neg Pred Value         0.7812   0.8025   1.0000
Prevalence             0.3464   0.3922   0.2614
Detection Rate         0.1634   0.2876   0.2614
Detection Prevalence   0.1634   0.4706   0.3660
Balanced Accuracy      0.7358   0.7161   0.9292
> install.packages("e1071")
> Housefile <- read.xlsx("/Users/nishkaranjan/Downloads/ProjectData.xlsx",1)
> summary(Housefile)
       X1              X2              X3              X4              X5
 Min.   :0.6200   Min.   :514.5   Min.   :245.0   Min.   :110.2   Min.   :3.50
 1st Qu.:0.6825   1st Qu.:606.4   1st Qu.:294.0   1st Qu.:140.9   1st Qu.:3.50
 Median :0.7500   Median :673.8   Median :318.5   Median :183.8   Median :5.25
 Mean   :0.7642   Mean   :671.7   Mean   :318.5   Mean   :176.6   Mean   :5.25
 3rd Qu.:0.8300   3rd Qu.:741.1   3rd Qu.:343.0   3rd Qu.:220.5   3rd Qu.:7.00
 Max.   :0.9800   Max.   :808.5   Max.   :416.5   Max.   :220.5   Max.   :7.00
 NA's   :528      NA's   :528     NA's   :528     NA's   :528     NA's   :528
       X6              X7              X8              Y1              Y2
 Min.   :2.00   Min.   :0.0000   Min.   :0.000   Min.   : 6.01   Min.   :10.90
 1st Qu.:2.75   1st Qu.:0.1000   1st Qu.:1.750   1st Qu.:12.99   1st Qu.:15.62
 Median :3.50   Median :0.2500   Median :3.000   Median :18.95   Median :22.08
 Mean   :3.50   Mean   :0.2344   Mean   :2.812   Mean   :22.31   Mean   :24.59
 3rd Qu.:4.25   3rd Qu.:0.4000   3rd Qu.:4.000   3rd Qu.:31.67   3rd Qu.:33.13
 Max.   :5.00   Max.   :0.4000   Max.   :5.000   Max.   :43.10   Max.   :48.03
 NA's   :528    NA's   :528      NA's   :528     NA's   :528     NA's   :528
   NA.
 Mode:logical
 NA's:1296




> pairs(~X1+X2+X3+X4+X5+X6+X7+X8, data = Housefile, main = "scatterplot")
> plot(Housefile$X1,Housefile)
Error in xy.coords(x, y, xlabel, ylabel, log) :
  'x' and 'y' lengths differ
> plot(Housefile$X1,Housefile$X2)
> plot(Housefile$X1,Housefile$X2, main = "Correlation b/w X1 and X2")
>  multiplemodel <- lm(formula = Y1~X1+X2+X3+X4+X5+X6+X7+X8, data =
Housefile)
> multiplemodel
```

```
Call:
lm(formula = Y1 ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8, data = Housefile)

Coefficients:
(Intercept)         X1         X2         X3         X4         X5         X6
   84.01342   -64.77343    -0.08729     0.06081         NA    4.16995    -0.02333
         X7         X8
   19.93274    0.20378

> summary(multiplemodel)

Call:
lm(formula = Y1 ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8, data = Housefile)

Residuals:
    Min      1Q  Median      3Q     Max
-9.8965 -1.3196 -0.0252  1.3532  7.7052
Coefficients: (1 not defined because of singularities)
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  84.013418  19.033613   4.414 1.16e-05 ***
X1          -64.773432  10.289448  -6.295 5.19e-10 ***
X2           -0.087289   0.017075  -5.112 4.04e-07 ***
X3            0.060813   0.006648   9.148  < 2e-16 ***
X4                  NA         NA      NA       NA
X5            4.169954   0.337990  12.338  < 2e-16 ***
X6           -0.023330   0.094705  -0.246  0.80548
X7           19.932736   0.813986  24.488  < 2e-16 ***
X8            0.203777   0.069918   2.915  0.00367 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 2.934 on 760 degrees of freedom
  (528 observations deleted due to missingness)
Multiple R-squared:  0.9162,  Adjusted R-squared:  0.9154
F-statistic:  1187 on 7 and 760 DF,  p-value: < 2.2e-16
> multiplemodel2 <- lm(formula = Y2~ X1+X2+X3+X4+X5+X6+X7+X8, data =
Housefile)
> multiplemodel2
Call:
lm(formula = Y2 ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8, data = Housefile)
Coefficients:
(Intercept)         X1         X2         X3         X4         X5         X6
   97.24575   -70.78771    -0.08824     0.04468         NA    4.28384     0.12151
         X7         X8
   14.71707    0.04070
```