USE CASE STUDY REPORT :
LIFE EXPECTENCY POST THORACIC SURGERY

Group No.: Group 08

Student Names: Tanmay Alsi and Meera Nagaria

I. Background and Introduction

Problem:
Often the term Thoracic Surgery is used interchangeably with Cardiothoracic Surgery, Adult Cardiac Surgery, Cardiovascular Surgery, Congenital Cardiothoracic Surgery, and General Thoracic Surgery. But for the layperson, Thoracic Surgery should be synonymous with General Thoracic Surgery.
About 80% of Thoracic Surgery involves surgery for some sort of cancer. This includes such tumors as lung cancer (which kills more people in the United States than colon, prostate and breast cancer combined), esophageal cancer, tumors of the chest wall (rib cage, sternum, etc.) and tumors of the mediastinum, or the space around the heart.

Provide background information to the use case study, including:

Goal:
We are trying to predict the post-operative live expectancy in lung cancer patients who undergo Thoracic Surgery. Doing so we can help hospitals to focus on patients who show low post-operative expectancy and detect a pattern to save their lives by checking the common attributes among low expectancy patients and directing research towards it.

Possible Solution:

- We will perform Exploratory Data analysis to analyze the data and check the correlation between the attributes and check if there any missing values and refine the data.
- Using Data visualization technique, we can determine the pattern in the data and decide on the machine learning model to be used.
- Use the Model to determine the post-operative death rate in 1 Year.

II. Data Exploration and Visualization

Brief Description:
The data was collected retrospectively at Wroclaw Thoracic Surgery Centre for patients who underwent major lung resections for primary lung cancer in the years 2007-2011. The Centre is associated with the Department of Thoracic Surgery of the Medical University of Wroclaw and Lower-Silesian Centre for Pulmonary Diseases, Poland, while the research database constitutes a part of the National Lung Cancer Registry, administered by the Institute of Tuberculosis and Pulmonary Diseases in Warsaw, Poland.
Source: UCI Machine Learning Repository.

DGN	Diagnosis - specific combination of ICD-10 codes for primary and secondary as well multiple tumours if any (DGN3,DGN2,DGN4,DGN6,DGN5,DGN8,DGN1)
PRE4	Forced vital capacity - FVC (numeric)
PRE5	Volume that has been exhaled at the end of the first second of forced expiration - FEV1 (numeric)
PRE6	Performance status - Zubrod scale (PRZ2,PRZ1,PRZ0)
PRE7	Pain before surgery (T,F)
PRE8	Haemoptysis before surgery (T,F)
PRE9	Dyspnoea before surgery (T,F)
PRE10	Cough before surgery (T,F)
PRE11	Weakness before surgery (T,F)
PRE14	T in clinical TNM - size of the original tumour, from OC11 (smallest) to OC14 (largest) (OC11,OC14,OC12,OC13)
PRE17	Type 2 DM - diabetes mellitus (T,F)
PRE19	MI up to 6 months (T,F)
PRE25	PAD - peripheral arterial diseases (T,F)
PRE30	Smoking (T,F)
PRE32	Asthma (T,F)
AGE	Age at surgery (numeric)
Risk1Y	1 year survival period - (T)rue value if died (T,F)

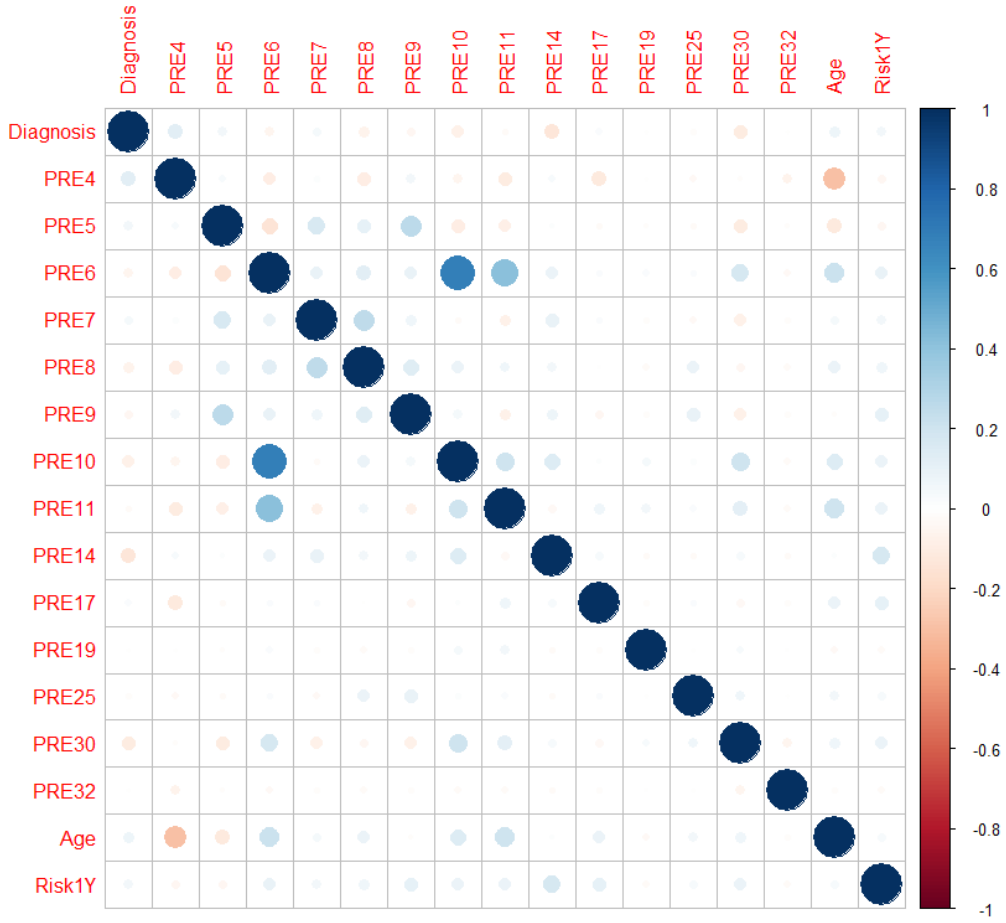
As We can see below the data has 470x17 dimensions i.e. it has 17 attributes with 470 instances.

```
> str(input)
tibble [470 x 17] (S3: tbl_df/tbl/data.frame)
 $ Diagnosis: num [1:470] 2 3 3 3 3 3 3 2 3 3 ...
 $ PRE4     : num [1:470] 2.88 3.4 2.76 3.68 2.44 2.48 4.36 3.19 3.16 2.32 ...
 $ PRE5     : num [1:470] 2.16 1.88 2.08 3.04 0.96 1.88 3.28 2.5 2.64 2.16 ...
 $ PRE6     : num [1:470] 1 0 1 0 2 1 1 1 2 1 ...
 $ PRE7     : num [1:470] 0 0 0 0 0 0 0 0 0 0 ...
 $ PRE8     : num [1:470] 0 0 0 0 1 0 0 0 0 0 ...
 $ PRE9     : num [1:470] 0 0 0 0 0 0 0 0 0 0 ...
 $ PRE10    : num [1:470] 1 0 1 0 1 1 1 1 1 1 ...
 $ PRE11    : num [1:470] 1 0 0 0 1 0 0 0 1 0 ...
 $ PRE14    : num [1:470] 4 3 1 1 1 1 2 1 1 1 ...
 $ PRE17    : num [1:470] 0 0 0 0 0 0 1 0 0 0 ...
 $ PRE19    : num [1:470] 0 0 0 0 0 0 0 0 0 0 ...
 $ PRE25    : num [1:470] 0 0 0 0 0 0 0 0 1 0 0 ...
 $ PRE30    : num [1:470] 1 1 1 0 1 0 1 1 1 1 ...
 $ PRE32    : num [1:470] 0 0 0 0 0 0 0 0 0 0 ...
 $ Age      : num [1:470] 60 51 59 54 73 51 59 66 68 54 ...
 $ Risk1Y   : num [1:470] 0 0 0 0 1 0 1 1 0 0 ...
```

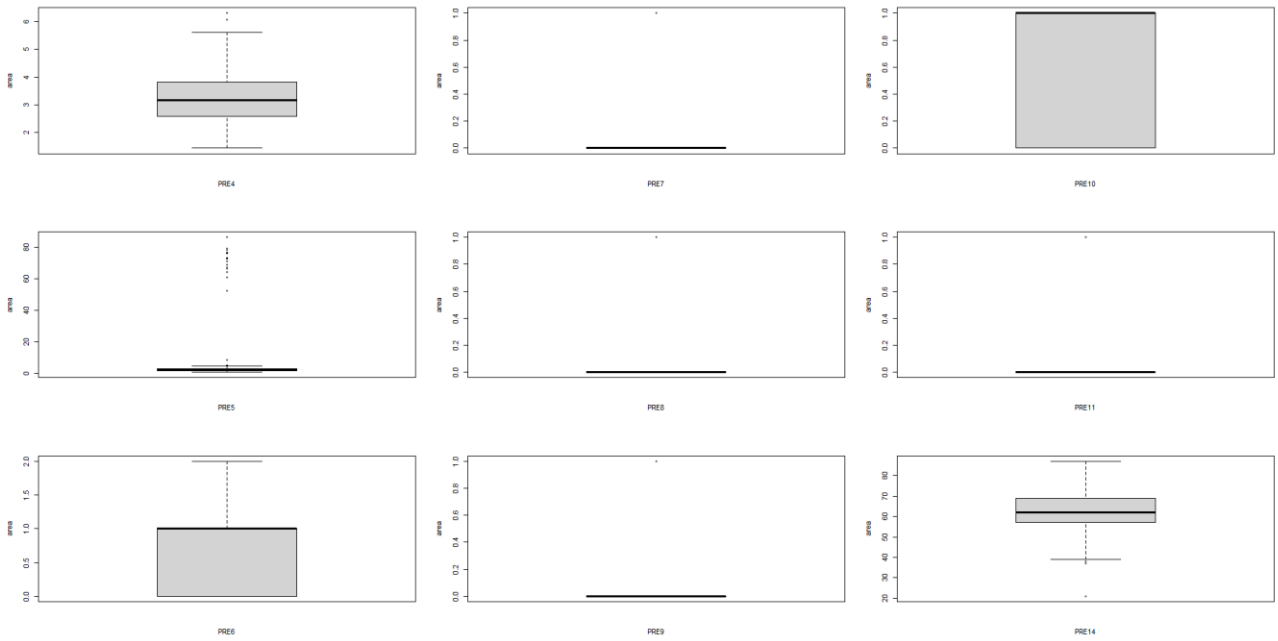
Visualization:

	Diagnosis	PRE4	PRE5	PRE6	PRE7	PRE8	PRE9	PRE10	PRE11	PRE14	PRE17	PRE19	PRE25	PRE30	PRE32	Age	Risk1Y
Diagnosis	1.00	0.12	0.06	-0.06	0.05	-0.06	-0.05	-0.08	-0.02	-0.13	0.03	-0.01	-0.02	-0.11	-0.01	0.08	0.06
PRE4	0.12	1.00	0.03	-0.09	0.02	-0.10	0.06	-0.05	-0.10	0.03	-0.12	-0.01	-0.04	-0.01	-0.06	-0.29	-0.05
PRE5	0.06	0.03	1.00	-0.14	0.16	0.10	0.26	-0.10	-0.09	0.01	-0.02	-0.01	-0.03	-0.10	-0.02	-0.12	-0.04
PRE6	-0.06	-0.09	-0.14	1.00	0.09	0.12	0.09	0.68	0.42	0.09	0.03	0.03	0.02	0.17	-0.03	0.21	0.09
PRE7	0.05	0.02	0.16	0.09	1.00	0.26	0.07	-0.02	-0.07	0.10	0.02	-0.02	-0.03	-0.08	-0.02	0.04	0.06
PRE8	-0.06	-0.10	0.10	0.12	0.26	1.00	0.13	0.08	0.06	0.06	0.00	-0.03	0.09	-0.04	-0.03	0.09	0.07
PRE9	-0.05	0.06	0.26	0.09	0.07	0.13	1.00	0.05	-0.07	0.07	-0.04	-0.02	0.10	-0.08	-0.02	-0.02	0.11
PRE10	-0.08	-0.05	-0.10	0.68	-0.02	0.08	0.05	1.00	0.20	0.14	0.02	0.04	0.02	0.20	-0.03	0.15	0.09
PRE11	-0.02	-0.10	-0.09	0.42	-0.07	0.06	-0.07	0.20	1.00	-0.04	0.07	0.06	0.03	0.12	-0.03	0.21	0.09
PRE14	-0.13	0.03	0.01	0.09	0.10	0.06	0.07	0.14	-0.04	1.00	0.04	-0.02	-0.02	0.04	-0.02	0.01	0.17
PRE17	0.03	-0.12	-0.02	0.03	0.02	0.00	-0.04	0.02	0.07	0.04	1.00	-0.02	0.03	-0.04	-0.02	0.09	0.11
PRE19	-0.01	-0.01	-0.01	0.03	-0.02	-0.03	-0.02	0.04	0.06	-0.02	-0.02	1.00	-0.01	0.03	0.00	-0.03	-0.03
PRE25	-0.02	-0.04	-0.03	0.02	-0.03	0.09	0.10	0.02	0.03	-0.02	0.03	-0.01	1.00	0.06	-0.01	0.06	0.04
PRE30	-0.11	-0.01	-0.10	0.17	-0.08	-0.04	-0.08	0.20	0.12	0.04	-0.04	0.03	0.06	1.00	-0.05	0.07	0.09
PRE32	-0.01	-0.06	-0.02	-0.03	-0.02	-0.03	-0.02	-0.03	-0.03	-0.02	-0.02	0.00	-0.01	-0.05	1.00	-0.02	-0.03
Age	0.08	-0.29	-0.12	0.21	0.04	0.09	-0.02	0.15	0.21	0.01	0.09	-0.03	0.06	0.07	-0.02	1.00	0.04
Risk1Y	0.06	-0.05	-0.04	0.09	0.06	0.07	0.11	0.09	0.09	0.17	0.11	-0.03	0.04	0.09	-0.03	0.04	1.00

We can observe from above table as well as the correlation plot below, there is significant correlation between the attribute PRE6 and PRE10 compared to other attributes.



We can observe from the boxplots below that the attribute, PRE 5 contributes to majority of the outliers in the dataset. The other attributes like AGE, PRE 4, PRE 7, PRE 11, PRE 14 consists of 1 or 2 outliers.

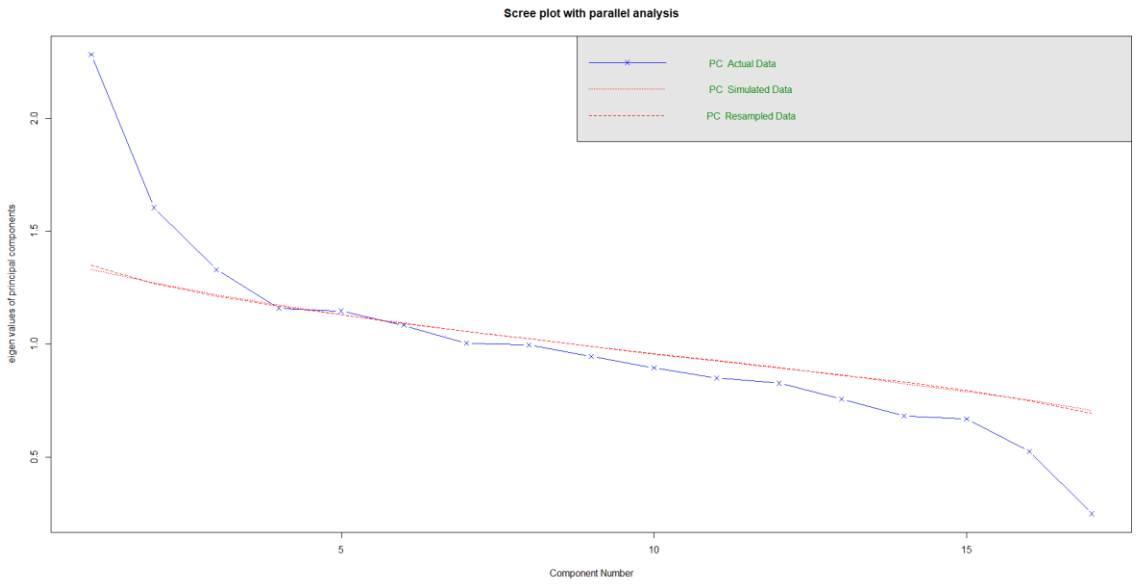


Missing Values:
There are no missing values in the dataset.

III. Data Preparation and Preprocessing

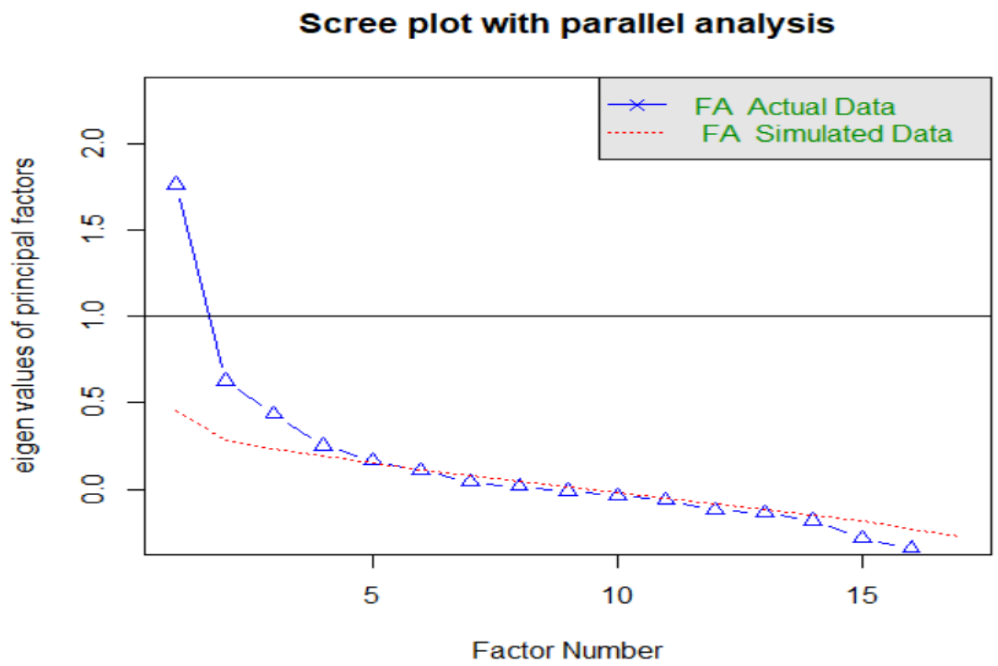
Initial Analysis of data showed that the columns in the data are stored as string values: “T” and “F”. Therefore, we converted them to int type: “1” and “0”. There are columns like DGN, PRE6 and PRE14 which have string value attached to the int. The string value does not provide any significance to the analysis. Thus, we removed the string value and only retained the int value. Similarly, ID column was also removed as it also does not provide any significant value to the analysis.

Principal Component Analysis:



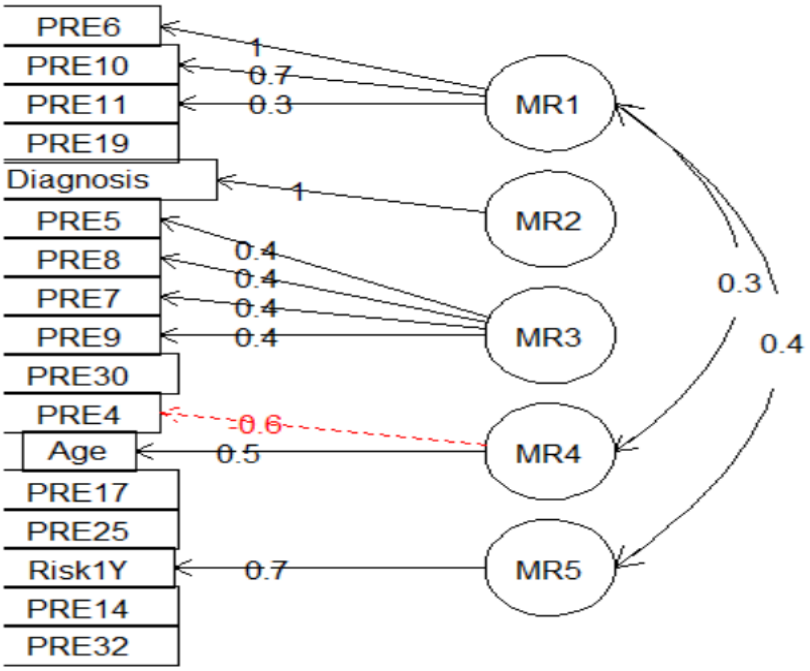
We can observe from the above plot suggests 3 components to be considered,
The 1st component contributes 12%, 2nd component 9% with the third component contributing 9% as well
The 1st component correlates strongly with 2 variables: PRE6 (Performance status - Zubrod scale (PRZ2,PRZ1,PRZ0) and PR10 (Cough before surgery (T,F)
The 2nd component has strong correlation with variables: PRE9(Dyspnoe b4 surgery), PRE7 (Pain before surgery) and PRE8(Haemoptysis b4 surgery)

Factor Analysis:



On performing Factorial method is used for Data reduction. It helps us to seek underlying unobserved (latent) variables that are reflected in the observed variables (manifest

variables). There are many different methods that can be used to conduct a factor analysis (such as principal axis factor, maximum likelihood, generalized least squares, unweighted least squares). In this scenario, there are total 5 factors are above the eigen value 1

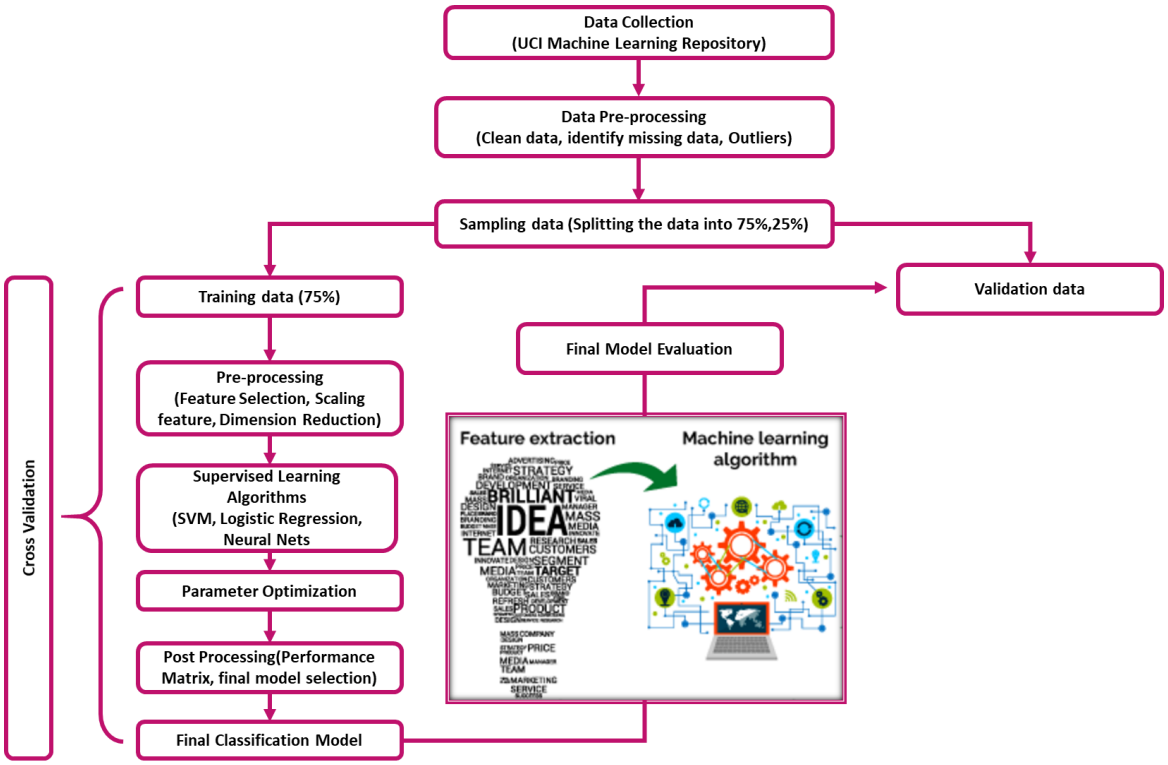


Factor 1 is strongly related with the variables : PRE6, PRE10, PRE11, Age-(Factor 4) are strongly related with the response class Risk1Y. PRE5, PRE7, PRE 8, PRE 9 are related to each other - (Factor 3).Factor 2 has only 1 variable relating to it- diagnosis. Also, age is inversely related with PRE4.

IV. Data Mining Techniques and Implementation

The past few years have seen machine learning has emerged as one of the trendiest topics within the technology sector as it allows computers to find hidden insights using multiple data mining techniques, from the volumes of data being collected without being explicitly programmed where to look. Data mining techniques are very effective and can be used for finding out insightful trends/results from the datasets. Data mining is also called as the analysis step for the knowledge Discovery in Database. The various data mining techniques are: Tracking patterns, Classification, Association, Outlier detection, Clustering, Regression and Prediction. The Data mining technique which we have used for our Case Study is Predictive classification: where we will be classifying one-year survival period of a patient who has undergone Thoracic surgery.
Among Predictive data mining techniques, Classification model is considered as the best-understood technique of all data mining approaches. The common characteristics of classification tasks are as supervised learning, categories dependent variable and assigning new data to one of a set of well-defined classes. Classification technique is used in customer segmentation, modeling businesses, credit analysis, and many other applications.

In a classification technique, you typically have historical data called labeled examples and new examples. Each labeled example consists of multiple predictor attributes (Age, Diagnosis, PRE4, PRE5, PRE6, PRE7, PRE8, PRE9, PRE10,.and so on) and one target attribute (Risk1Y) that is a class label. While unlabeled examples only consist of the predictor attributes. The goal of classification is to construct a model using the data from history and accurately predicts the new class of examples. So, in our classification technique, from the various independent variables we can predict the life expectancy (max up to one year) of a patient who is diagnosed with lung cancer and has undergone thoracic surgery. Classification task begins with build data in database also known as training data for which the target values are known. There are different classification algorithms available that uses their different techniques for finding relations between the predictor attributes values and the target values (Yes, No) in the build data. After getting the targeted data, these relations are summarized in a model, so that they can be applied to new cases further with unknown target values for predicting target values. The workflow of the predictive classification model is as shown below:



You are expected to explore multiple data mining techniques as appropriate to your problem. Clearly state the problem in data mining context (e.g., classification, prediction, supervised/unsupervised learning, etc.). It is desirable to have a flowchart for the entire process from data cleaning/manipulation/variable selection and transformation to specific techniques/algorithms implemented in R.

V. Performance Evaluation

With different data mining techniques and various predictive classification algorithms, these algorithms have been applied on different dataset to find out the efficiency of the algorithm and improve the performance by applying data preprocessing techniques and feature selection and prediction of new class labels. Performance evaluation is basically done to evaluate how well the model is functioning and to review if the model is serving the purpose of the case study. On performing various Supervised machine learning algorithms, we then calculated various measures to evaluate the learning model and found the rate of life expectancy of an individual diagnosed with lung cancer.

To evaluate the performance of the model, we generally divide the data set into training and validation set comprising of 75% and 25% respectively.

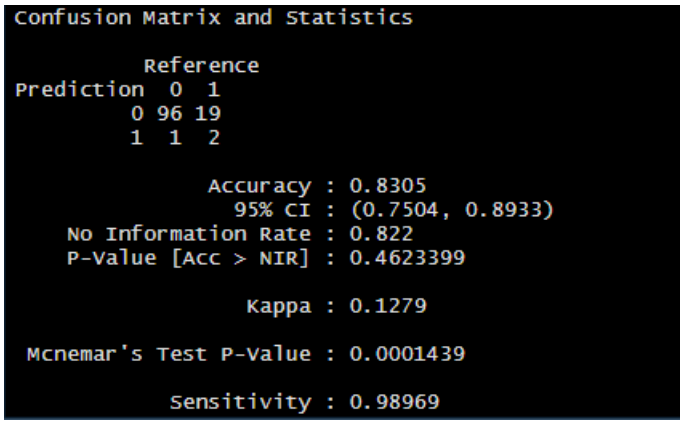
Accuracy for different Algorithms.
Logistic Regression: 83%
SVM: 78.8%
Neural Nets: 63%

We observe that the accuracy for Neural Networks is significantly lower than that of the other two. This is because the dataset size is small. SVM does better generalization with respect to smaller datasets.

We will further compare the performance of SVM and Logistic Regression using confusion matrix.
SVM:

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 93 21
         1  4  0

               Accuracy : 0.7881
                 95% CI : (0.7033, 0.858)
    No Information Rate : 0.822
    P-Value [Acc > NIR] : 0.859942

                  Kappa : -0.0604

 Mcnemar's Test P-Value : 0.001374

            Sensitivity : 0.9588
```

Logistic Regression:



```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 96 19
         1  1  2

               Accuracy : 0.8305
                 95% CI : (0.7504, 0.8933)
    No Information Rate : 0.822
    P-Value [Acc > NIR] : 0.4623399

                  Kappa : 0.1279

 Mcnemar's Test P-Value : 0.0001439

            Sensitivity : 0.98969
```

We can see that logistic regression and SVM performs almost same with logistic regression performing slightly better with 0.9896 sensitivity as opposed to SVM's 0.9588

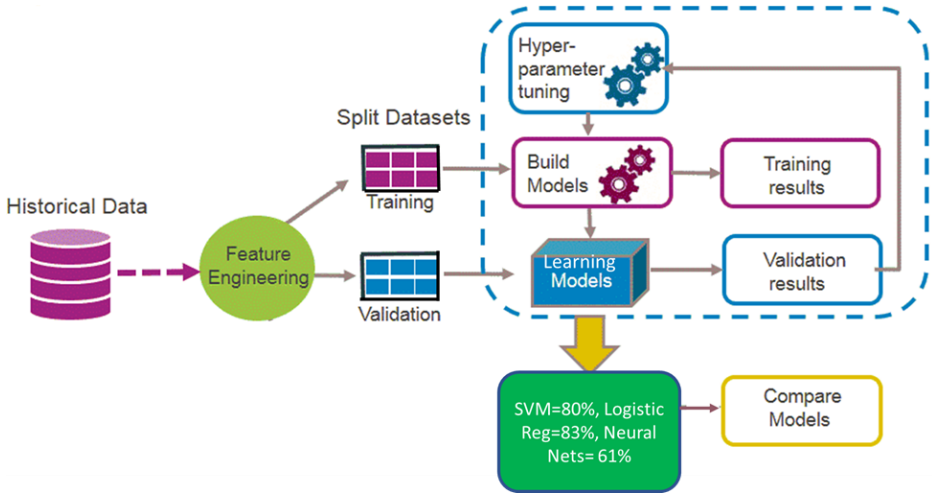
VI. Discussion and Recommendation

The logistic Regression model excelled the most for our problem with highest accuracy. The Logistic Regression works best with binary variables. Our data mostly consists of binary variable including the response variable.
Thus, we recommend using Logistic Regression for this project.

Unfortunately, the amount of data on the patients available on UCI repository was very limited. Moreover, we see some attributed with low correlation and overlapping which make it hard for the model to distinguish the differences.
We can use boosting techniques to further optimize the models. Furthermore, we can collect data from multiple sources and hospital and use a python script to consolidate the data from multiple sources to create one large dataset for our model training.

The study we performed has great applications in the medical field. We are predicting the mortality rate for a surgery which can be used as risk measure for patients who are about to undergo the surgery.
As mentioned above the primary challenge with this project was the limited data as it was taken from a specific study performed on limited number of patients.

VII. Summary



We followed the process displayed in the above flow chart.

1. We searched for dataset on public domain: UCI Repository.
2. Data Exploration: Performed analysis on the data to get an overview about the dataset.
3. Data Cleaning: Checked for outliers and missing values. Normalized the data. Performed regression to find most important predictors.
4. Data visualization: Correlation plots, Box plots and PCA and FA graphs.
5. Data splitting: Dividing data into Training and Validation datasets.
6. Performing Evaluations: Accuracy, Confusion Matrix.
7. Conclusion: Selection of best predicting Model. – Logistic Regression.

This case study provided an insight into the application of machine learning in health informatics. Moreover, use of multiple machine learning algorithms taught us that different algorithms excel at different dataset. For this case study Logistic Regression might be proven best but this might be different case with different data, like data with continuous value or very high variance.

Appendix: R Code for use case study

```
#Libraries used
library(readxl)
library(neuralnet)
library(psych)
library(dplyr)
library(car)
library(MASS)
library(gvlma)
library(caret)
library(gains)
library(ggplot2)
library(rpart)
library(rpart.plot)
library(e1071)
library(GGally)
library(gmodels)
library(class)

#path of the input dataset
setwd("C:/Users/meera/OneDrive - Northeastern University/SEM 2/Data mining/Case
study")
input<- read_excel("Input data.xlsx",sheet="Sheet1")
str(input)
head(input)

# PART 1 - data pre-processing
input <- input %>% mutate(Diagnosis = case_when(
  Diagnosis == 'DGN1' ~ 1,
  Diagnosis == 'DGN2' ~ 2,
  Diagnosis == 'DGN3' ~ 3,
  Diagnosis == 'DGN4' ~ 4,
  Diagnosis == 'DGN5' ~ 5,
  Diagnosis == 'DGN6' ~ 6,
  Diagnosis == 'DGN8' ~ 8))
input

input <- input %>% mutate(PRE6 = case_when(
  PRE6 == 'PRZ0' ~ 0,
  PRE6 == 'PRZ1' ~ 1,
  PRE6 == 'PRZ2' ~ 2))
input

input <- input %>% mutate(PRE7 = case_when(
```

```
  PRE7 == 'F' ~ 0,
  PRE7 == 'T' ~ 1))
input

input <- input %>% mutate(PRE8 = case_when(
  PRE8 == 'F' ~ 0,
  PRE8 == 'T' ~ 1))
input

input <- input %>% mutate(PRE9 = case_when(
  PRE9 == 'F' ~ 0,
  PRE9 == 'T' ~ 1))
input

input <- input %>% mutate(PRE10 = case_when(
  PRE10 == 'F' ~ 0,
  PRE10 == 'T' ~ 1))
input

input <- input %>% mutate(PRE11 = case_when(
  PRE11 == 'F' ~ 0,
  PRE11 == 'T' ~ 1))
input

input <- input %>% mutate(PRE14 = case_when(
  PRE14 == 'OC11' ~ 1,
  PRE14 == 'OC12' ~ 2,
  PRE14 == 'OC13' ~ 3,
  PRE14 == 'OC14' ~ 4))
input

input <- input %>% mutate(PRE17 = case_when(
  PRE17 == 'F' ~ 0,
  PRE17 == 'T' ~ 1))
input

input <- input %>% mutate(PRE19 = case_when(
  PRE19 == 'F' ~ 0,
  PRE19 == 'T' ~ 1))
input

input <- input %>% mutate(PRE25 = case_when(
  PRE25 == 'F' ~ 0,
  PRE25 == 'T' ~ 1))
input
```

```
input <- input %>% mutate(PRE30 = case_when(
  PRE30 == 'F' ~ 0,
  PRE30 == 'T' ~ 1))
input

input <- input %>% mutate(PRE32 = case_when(
  PRE32 == 'F' ~ 0,
  PRE32 == 'T' ~ 1))
input

input <- input %>% mutate(Risk1Y = case_when(
  Risk1Y == 'F' ~ 0,
  Risk1Y == 'T' ~ 1))

head(input)
str(input)

#Part 2- Principal Component Analysis and Factor Analysis

#Principal Component Analysis
plot<-fa.parallel(input,fa='pc',n.iter=100,show.legend=TRUE,main="Scree plot with
parallel analysis")

input_analysis<-principal(input,nfactors=3,rotate = "promax",score=TRUE)
input_analysis

head(input_analysis$scores)

graph<-factor.plot(input_analysis,labels =rownames(input_analysis$loadings) )

#Interpretation : 3- Principal components, The 1st component contributes 12%, 2nd
component- 9% and third component contributes again 9%.
# The 1st component has basically correlates most strongly with 2 variables: PR6
(Performance status - Zubrod scale (PRZ2,PRZ1,PRZ0)
#and PR10 (Cough before surgery (T,F)
#The 2nd component has strong correlation with variables: PR9(Dyspnoe b4 surgery),
PR7 (Pain before surgery) and PR8(Haemoptysis b4 surgery)

#Factor Analysis
fa.parallel(input,n.obs=470,fa="fa",n.iter=100,show.legend=TRUE, main="Scree plot
with parallel analysis")

#rotate the factors
input_varimax<-fa(input, nfactors=5,rotate = "varimax")

#compute scores
```

```
head(factor.scores(input,input_varimax))

#orthogonal solution
factor.plot(input_varimax)

#oblique soution
input_promax<-fa(input, nfactors=5,rotate = "Promax")
fa.diagram(input_promax, simple=FALSE)

#Interpretation : In total 5 factors are above the eigen value 1. Factor 1 is strongly related with the variabes : PRE6, PRE10, PRE11, Age-(Factor 4) are strongled related with the reponse class Risk1Y.
#PRE5, PRE7, PRE 8, PRE 9 are related to ech other - (Factor 3).Factor 2 has only 1 variable relating to it- diagnosis. Also Age is inversly related with PRE4.

#PART 3 - DATA VIZUALIZATION

correlation <- cor(input, method = "pearson", use = "complete.obs")
round(correlation,2)
corrplot(correlation)
par(mfcol = c(3,3))
boxplot(input$PRE4, xlab = "PRE4", ylab = "area")
boxplot(input$PRE5, xlab = "PRE5", ylab = "area")
boxplot(input$PRE6, xlab = "PRE6", ylab = "area")
boxplot(input$PRE7, xlab = "PRE7", ylab = "area")
boxplot(input$PRE8, xlab = "PRE8", ylab = "area")
boxplot(input$PRE9, xlab = "PRE9", ylab = "area")
boxplot(input$PRE10, xlab = "PRE10", ylab = "area")
boxplot(input$PRE11, xlab = "PRE11", ylab = "area")
boxplot(input$Age, xlab = "PRE14", ylab = "area")



#PART 4 - DIMENSION REDUCTION
str(input)

Regression_t1 <- rpart(Risk1Y ~ ., data= input,method= "anova", control =
rpart.control(maxdepth = 3))

printcp(Regression_t1)

summary(Regression_t1)
#Using Regression Algorithm, we found the Variable Importance for the best performance of the algorithm.
#Top 7 most Imp vars are: Diagnosis, PRE4, PRE5, PRE6, PRE14, PRE30 and Age.
```

```
#PART 5 - ALGORITHM IMPLEMENTATION
#(a): Support Vector Machine
input_svm<-input

str(input_svm)

training_data=sample(row.names(input_svm), dim(input_svm)[1]*0.75)
validation_data=setdiff(row.names(input_svm), training_data)

train_df =as.data.frame (input_svm[training_data, ])
valid_df = as.data.frame( input_svm[validation_data, ])

svm_model <- svm(Risk1Y ~ ., data=train_df, cost=100, gamma =1,trControl =
trainControl(method = "cv"), method = "svmPoly")
summary(svm_model)

pred <- predict(svm_model,valid_df[,-17])

table(pred,valid_df$Risk1Y)

data.frame(actual = valid_df$Risk1Y[1:5], predicted = pred[1:5])
#RMSEsvm <- rmse(actual,Predicted)
fitted.results.svm <- ifelse(pred > 0.5,1,0)

fitted.results.svm<-as.factor(fitted.results.svm)

cm<-confusionMatrix(data=fitted.results.svm,reference=as.factor(valid_df$Risk1Y))
Accuracy_svm<-round(cm$overall[1],2)
Accuracy_svm
#Accuracy -78.83%

#(b)Logistic Regression

set.seed(1234)
pred_logit <- glm(Risk1Y ~ ., data = train_df, family =binomial)
summary(pred_logit)

coef(pred_logit)

exp(coef(pred_logit))

logit.reg.pred<-predict(pred_logit, newdata = valid_df[,-17], type = "response")

data.frame(actual = valid_df$Risk1Y[1:5], predicted = logit.reg.pred[1:5])
```

```
fitted.results.cat <- ifelse(logit.reg.pred > 0.5,1,0)

require(caret)
fitted.results.cat<-as.factor(fitted.results.cat)
cm<-confusionMatrix(data=fitted.results.cat,reference=as.factor(valid_df$Risk1Y))

Accuracy_logistic<-round(cm$overall[1],2)
Accuracy_logistic
#Accuracy – 83%

#(c)NeuralNets
input_nn<-input[]
str(input_nn)

#Normalize the data/Scale the data
max = apply(input_nn , 2 , max)
min = apply(input_nn, 2 , min)
scaled_nn = as.data.frame(scale(input_nn, center = min, scale = max - min))

str(scaled_nn)

#Dividing the dataset into training and validation
training_data=sample(row.names(scaled_nn), dim(scaled_nn)[1]*0.75)
validation_data=setdiff(row.names(scaled_nn), training_data)

train_df =as.data.frame (scaled_nn[training_data, ])
valid_df = as.data.frame( scaled_nn[validation_data, ])

#Applying Nueral Nets Algo
nn_impvar <- neuralnet(Risk1Y ~ ., data = train_df, hidden = 1, threshold = 0.01,
linear.output = T, algorithm = "rprop+",learningrate = 0.1, stepmax = 1e7)

nn_impvar
plot(nn_impvar)
nn_impvar$result.matrix
nn.results <- neuralnet::compute(nn_impvar, valid_df[,-17])
results <- data.frame(actual = valid_df$Risk1Y, prediction = nn.results$net.result)
nn.results <- (nn.results$net.result * (max(valid_df$Risk1Y) - min(valid_df$Risk1Y))) +
min(valid_df$Risk1Y)
RMSE.nn_valid = (sum((valid_df$Risk1Y - nn.results)^2) / nrow(valid_df)) ^ 0.5
print(1-RMSE.nn_valid)
#Accuracy -62%

#plot(valid_df$Risk1Y, nn.results, col='blue', pch=16, ylab = "predicted rating NN", xlab
= "real rating")
#abline(0,1)
```