Breast Cancer Classification

Tram Ho

MA 5790

October 2023

**Abstract:**

According to WebMD, the medical bill for one treatment of breast cancer stage 0 is $48,477. The bill rose to about $71,909 after two years. For stage I and II, the range of the treatment bill is about $61,621-$97,066. The bill range is so high and also it is dangerous for women diagnosed with cancer. The cancer can go back after the full treatment. The bills go up after every treatment. And more importantly, sometimes, it costs women their lives.

One of the treatments is chemotherapy. And the costs range from $10,000 to $ 100,000. It is very painful for women who undergoes chemotherapy. My friend’s mom said she rather die than go for more chemotherapy treatments. And it put me in thinking of how to prevent the women breast cancers or at least find the way to diagnosis it so we can treat the patient at their earlier stages of breast cancer. These purposes, I really want to do in the future. In this project the specific purpose is predicting in advance the recurrence of breast cancer.

The purpose of this project is to predict if the recurrence of cancer will occur again in women who had been treated before. And the analyzing was done by running some classification models to find the best optimal model.

**Table of Contents:**

Abstract: ……………………………………………………………………………………2

1. Background: ……………………………………………………………………4
2. Variable Introduction and Definitions: ………………………………………4
3. Analyzing Data: ………………………………………………………………5
4. Preprocessing of the predictors: ……………………………………………7
5. Splitting Data: …………………………………………………………………8
6. Model Tuning: …………………………………………………………………8

* Important Variables of support Vector Machine Models: ………………
* Graphs of Accuracy and ROC of models in this project: ………………

1. Code : …………………………………………………………………………13
2. Resource: ……………………………………………………………………..25

**1 Background**: One of the main causes of death is cancer and breast cancer is the most common cancer in women. Breast cancer can cause bankruptcy financially and emotionally to a family.

**2 Variable Introduction and Definitions:**

The data set was created by the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. It was put in the Kaggle website. The data set was obtained from October, 1988 until the present time and is updating when there is new data. The last time it was updated is August, 2023.

The data set has 286 observations.

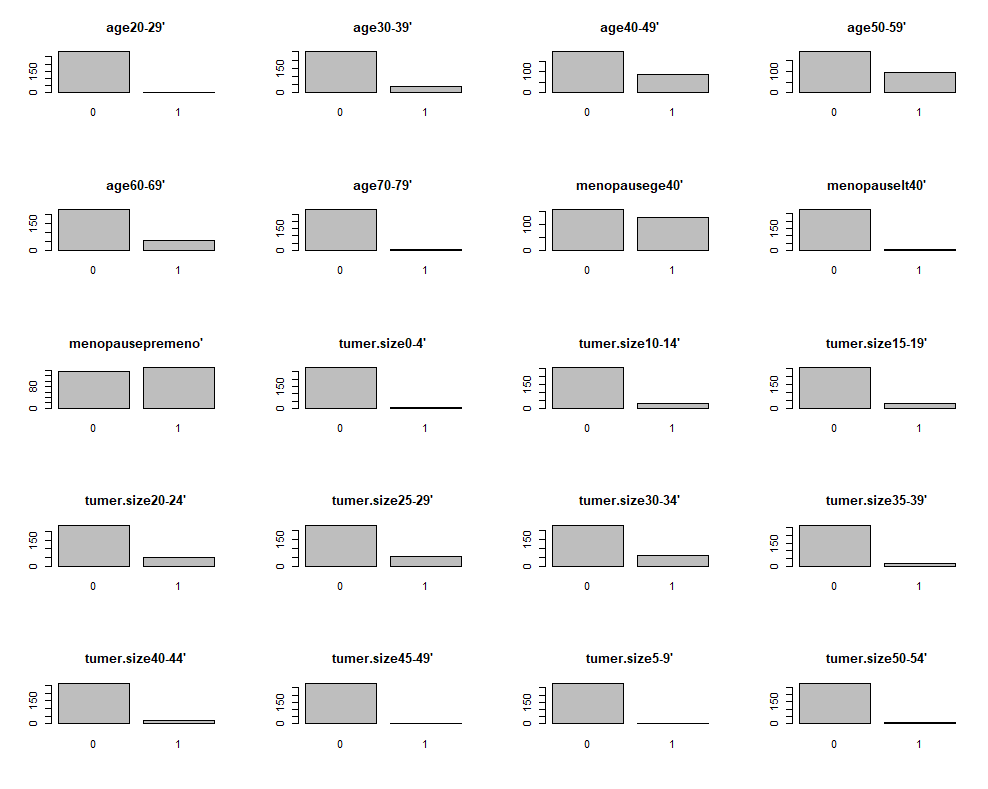
There are 10 categorical variables in the data set. Below is the description of variables.

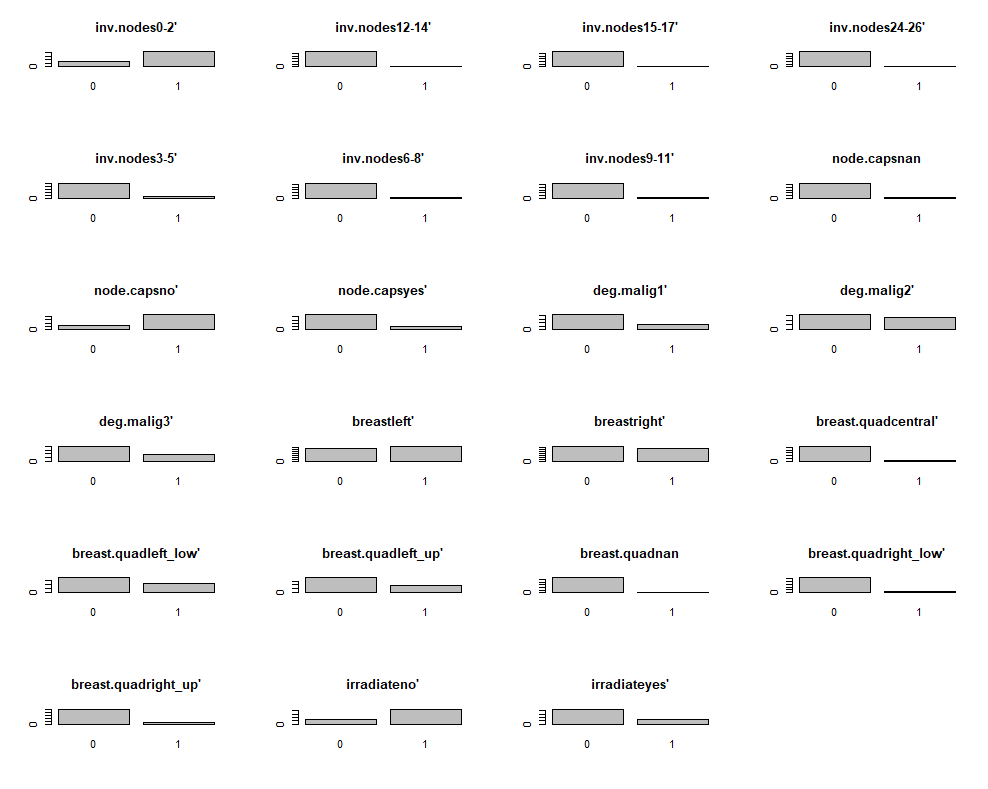
1. Class: The response variable classifies if there is recurrence of cancer or not. (no-recurrence-events, recurrence-events)
2. Age: Patient’s age range. These ranges of age are : 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-99.
3. Menopause: The natural biological process that marks the end of a woman’s menstrual cycle. It includes: lt40, ge40, premeno.
4. Tumor-size: Tumor’s dimension or measurement. Tumor size are: 0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59.
5. Inv-nodes: The axillary lymph nodes number that contain metastatic breast cancer visible on 0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35, 36-39.
6. Node-caps: This variable indicates whether the cancer does metastasise to a lymph node. These are yes or no.
7. Deg-malig: 1, 2, 3.
8. Breast: The breast variable indicates the position of breasts. They are left or right.
9. Breast-quad: The breasts are divided into 4 quadrants with the nipple as the center. These are: left-up, left-low, right-up, right-low, central.
10. Irradiate: The irradiation is known as the radiation therapy. It is a common treatment for breast cancer. The variable indicates whether there is the irradiation for the patient or not. (yes, no)

**3 Analyzing Data**

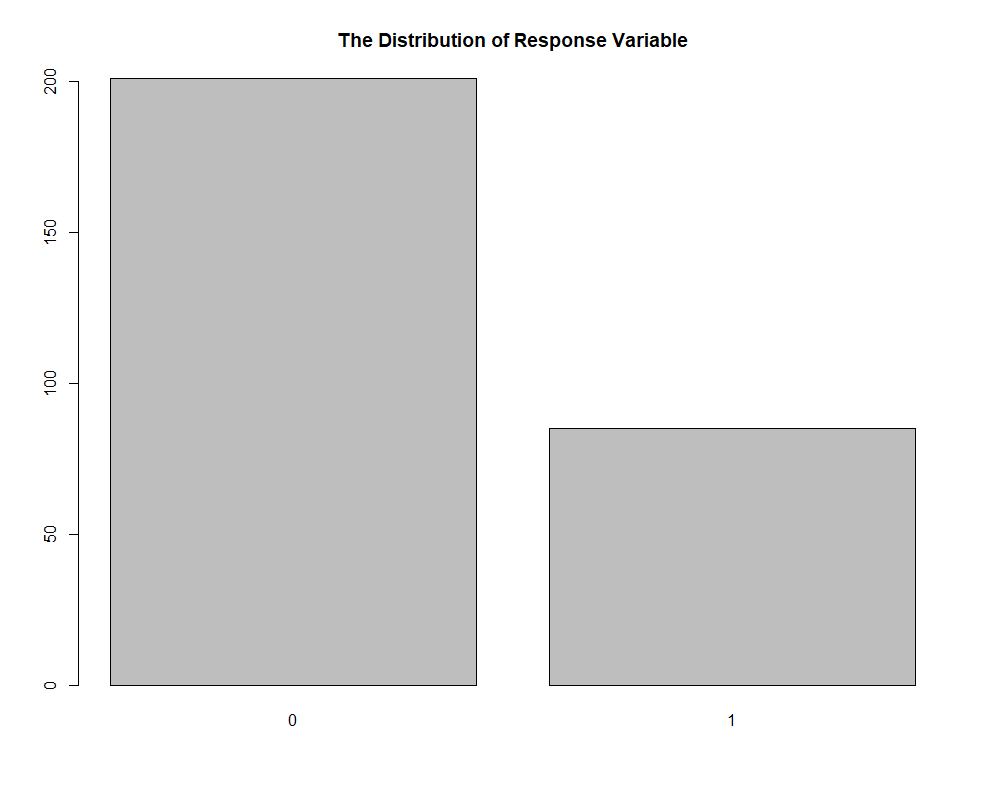
Since all the predictors are categorical ones, so dummy variables were created before preprocessing data. The response variable was classified into two levels. Zero (0) as no recurrence event and one (1) as recurrence-event.

Then I graphed the categorical predictors to have a general look of the distribution of data.



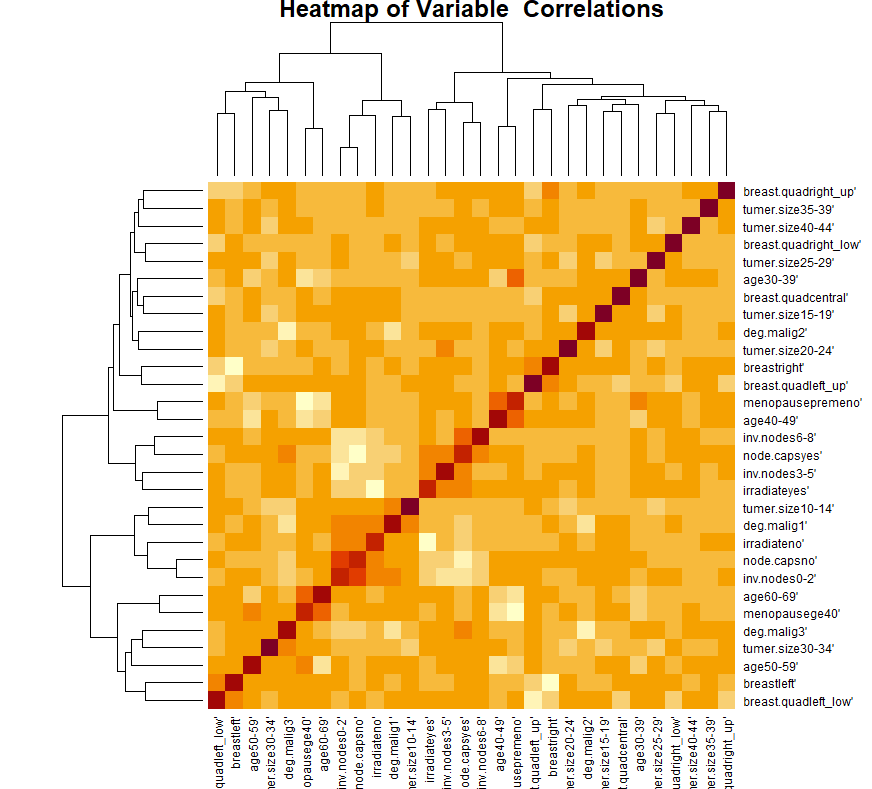


The below graph is the distribution of response variable:



**4 Preprocessing of the predictors**

The near zero variance variables were removed. There are thirteen of them. Since there is missing value in the data set so Imputing function kNN was used to impute data. Then the high correlated variables were removed with the cut-off as 0.8. There were 4 highly correlated variables in the data set. Below is the heatmap of variable correlations before removing highly correlated variables.



Since the skewness of some variables are out of normal range so Boxcox was used and because the data was measured in different unit so center and scale methods were used.

Since all of the data variables are categorical so there are not any outliers in the data. However, after using BoxCox, Center and Scale methods the skewnesses of data are still out of normal range so the spartialSign function was applied to data.

**5 Splitting Data:**

After dummy transformation. The response variable contains 0 as no recurrence event and 1 as recurrence event. The response variable distribution was summarized as in the below table:

|  |  |
| --- | --- |
| Response type | Number of observations |
| 0 | 201 |
| 1 | 85 |

The response of data set was splittted by using createDataPartiontion function. Then the result was used to subset the data to create the training set and the data set. The data set has 286 observations. After splitting, the training set has 229 observations and the test set has 57 observations.

**6 Model Tuning**

The models were tuned using cross validation. The following table summarized the important metric of models.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Specificity | Sensitivity | Accuracy | Kappa | Area under the Curve |
| Linear Discriminant Analysis | 0.3553 | 0.7949 | 0.6614 | 0.1579 | 0.7818 |
| Partial Least Square Discriminant Analysis | 0.3386 | 0.8314 | 0.6818 | 0.1843 | 0.7592 |
| Logistic Model | 0.3812 | 0.8174 | 0.685 | 0.2105 | 0.7845 |
| Penalized Model | 0.1450 | 0.9476 | 0.7039 | 0.1167 | 0.7431 |
| Support Vector Machine | 0.2571 | 1.0000 | 0.7729 | 0.3246 | 0.8382 |
| Flexible Discriminant Analysis | 0.5333 | 0.9286 | 0.8246 | 0.5052 | 0.6635 |
| Neural Network | 0.06667 | 1.00000 | 0.7544 | 0.0952 | 0.8078 |
| K-Nearest Neighbors | 0.1000 | 0.9811 | 0.7118 | 0.1067 | 0.7162 |
| Naïve Bayes | 0.3333 | 0.9286 | 0.7719 | 0.3081 | 0.8078 |

Based on area under the curve, two best model are Support Vector Machine Model and Naïve Bayes Model. Below table are the Kappa and Area under the Curve of these two models on the test set.

|  |  |  |
| --- | --- | --- |
| Top Two Models | | |
| Model | Kappa | Area under the Curve |
| Support Vector Machine | 0.1086 | 0.6054 |
| Naïve Bayes | 0.0452 | 0.6694 |

|  |  |  |
| --- | --- | --- |
| Confusion Matrix of the Support Vector Machine on Test Set | | |
|  | Reference | |
| Prediction | 0 | 1 |
| 0 | 39 | 11 |
| 1 | 3 | 4 |

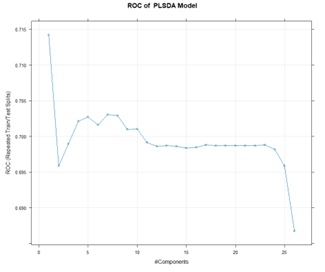
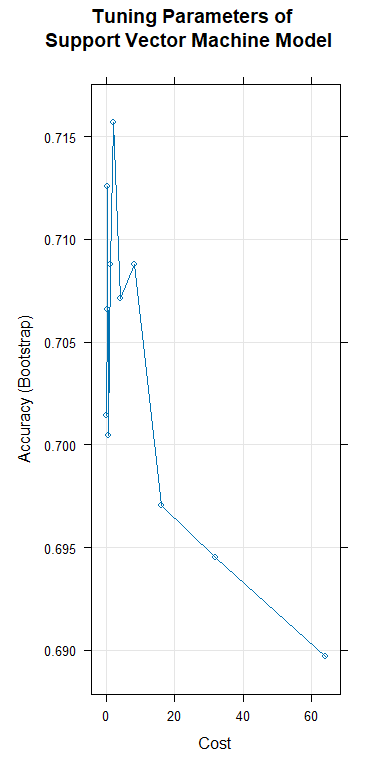
The support vector machine misclassified 11 patients who had recurrence breast cancer as no recurrence cases. It also misclassified 3 patients who had no recurrence breast cancer as having breast cancer. Misclassifying 11 patients who had cancer as no cancer is a high misclassifying number. I think one of the reasons for this is the small size of the data set. The model does not have a chance to learn from a big size data set. Since it is very important to predict accurately who will have breast cancer coming back, this model is still not the one as I desired. Even though this is the best model for this project, I think we need to find the way to improve the accuracy of prediction. One thing I expect is that if we have more data, we will run the model again to see if we can improve the accuracy of model.

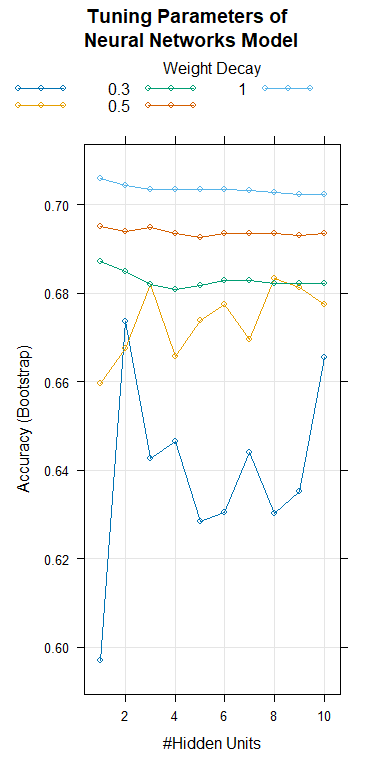
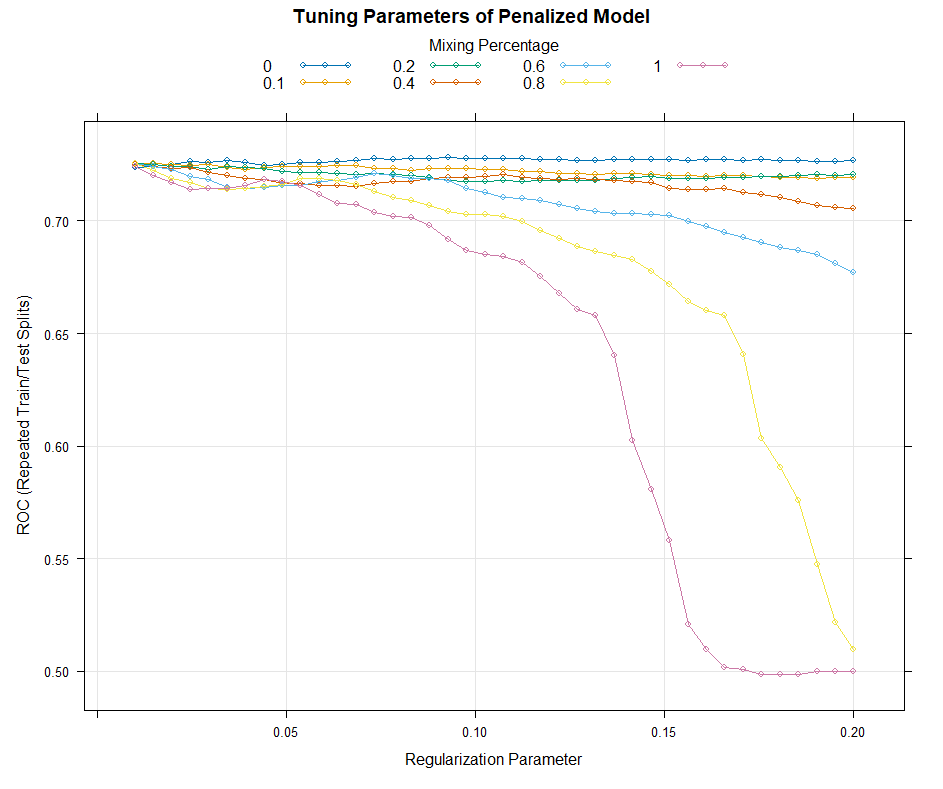
**7 Important Variables of support Vector Machine Models**

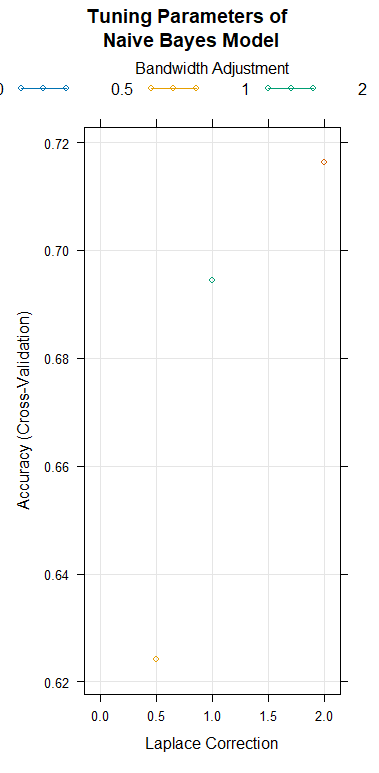
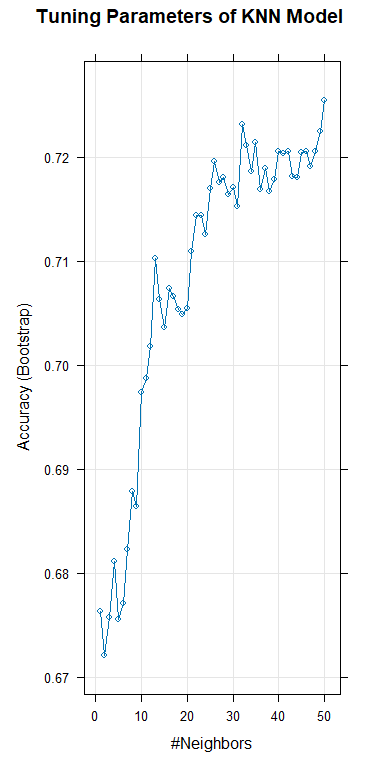
|  |  |
| --- | --- |
| Variable | Importance |
| deg.malig3' | 100.00 |
| inv.nodes0-2' | 99.60 |
| node.capsyes' | 97.40 |
| breast.quadright\_up' | 90.28 |
| inv.nodes6-8' | 79.69 |

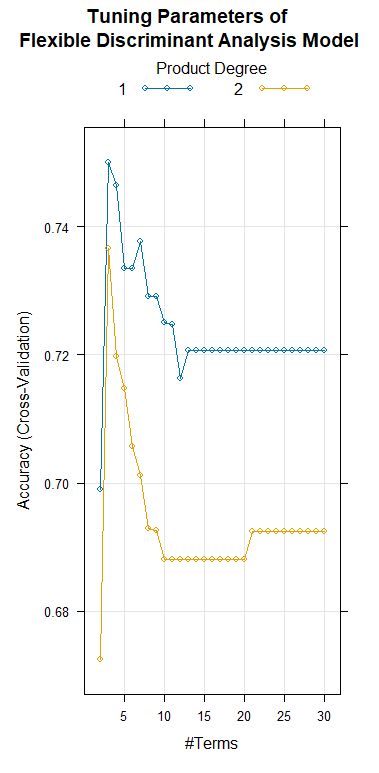
The first important variable is deg.malg 3. I think this is make sense because of the high degree of malignant so it must be important in the model. The second and the fifth come from the same variable with different levels. And the tumor size which I thought would be the important one before running models turns out not in the first five important variables.

Graphs of Accuracy and ROC of models in this project:









**Code**

library(e1071)

library(caret)

library(recipes)

library(VIM)

library(MLmetrics)

library(pROC)

library(mda)

library(kernlab)

library(pamr)

library(klaR)

# Reading Data

data<-read.csv("C:/Users/tramh/ Documents/MA 5790 Predictive Statistic/Project 2/breast-cancer-data.csv")

# Creating dummy Variables

simple<-dummyVars(class~age+menopause+tumer.size+inv.nodes+node.caps+deg.malig+breast+breast.quad+irradiate,data=data)

data1<-predict(simple,data)

dim(data1) #286 43

unique(data$class)

y<-c()

y[which(data$class=="recurrence-events'") ]=1

y[which(data$class=="no-recurrence-events'") ]=0

#Analysing and graph data

# Barplots of categorical predictors

names<-colnames(data1)

par(mfrow=c(5,4))

for (i in 1:20){

counts<-table(data1[,i])

pl<-barplot(counts,main=names[i])

}

par(mfrow=c(6,4))

for (i in 21:43){

counts<-table(data1[,i])

pl<-barplot(counts,main=names[i])

}

# Barplot of response

tableY<-table(y)

barplot(tableY, main="The Distribution of Response Variable")

#Preprocessing Data

heatmap(cor(data1), main="Heatmap of Variable Correlations \n before Transformation")

apply(data1,2 ,skewness)

n<-nearZeroVar(data1,freqCut=95/5)

data2<-data1[,-n]

dim(data2)

trans1<-kNN(data2,imp\_var=FALSE)

corre<-cor(trans1)

heatmap(corre, main="Heatmap of Variable Correlations")

t<-findCorrelation(corre,cutoff=0.8)

data3<-trans1[,-t]

apply(data3,2 ,skewness)

trans<-preProcess(data3,method=c("BoxCox","center","scale"))

data4<-predict(trans,data3)

apply(data4,2 ,skewness)

data5<-spatialSign(data4)

apply(data5,2 ,skewness)

heatmap(cor(data5), main="Heatmap of Correlations \n after Transformation")

**#Splititng Data**

y1<-c()

y1[which(y==0)]<-"Class1"

y1[which(y==1)]<-"Class2"

unique(y1)

y2<-factor(y1,labels=c('Class1','Class2'))

set.seed(123)

index<-createDataPartition(y,p=0.8,list=FALSE)

trainX<-data5[index,]

trainY<-y2[index]

testX<-data5[-index,]

testY<-y2[-index]

length(trainY) #229

length(testY) # 57

dim(trainX) #229 26

dim(testX) #57 26

**# Buiding Model**

**# Logistic Regression**

set.seed(1)

ctrl <- trainControl(method = "LGOCV",

summaryFunction = twoClassSummary,

classProbs = TRUE,

##index = list(simulatedTest[,1:4]),

savePredictions = TRUE)

lModel <- train(trainX,

y = trainY,

method = "glm",

metric = "auc",

# preProc=c("center","scale","pca")

trControl = ctrl)

#Predict on the training set

LPred<-predict(lModel, trainX)

postResample(pred=LPred,obs=trainY)

confusionMatrix(lModel$pred$pred,lModel$pred$obs)

LPredProbs<-predict(lModel, newdata=trainX,type="prob")

LFullRoc<-multiclass.roc(response=trainY, predictor=LPredProbs)

auc(LFullRoc)# 0.8149

**#LDA**

```{r,echo=FALSE}

ctrl <- trainControl(method = "LGOCV",

summaryFunction =multiClassSummary,

classProbs = TRUE,

##index = list(simulatedTest[,1:4]),

# preprocess("")

savePredictions = TRUE)

LDATrain <- train(x=trainX,trainY,method = "lda", metric = "auc", trControl = ctrl)

#Predict on the training set

LPred<-predict(LDATrain, trainX)

postResample(pred=LPred,obs=trainY)

confusionMatrix(LDATrain$pred$pred,LDATrain$pred$obs)

LPredProbs<-predict(LDATrain, newdata=trainX,type="prob")

FullRoc<-multiclass.roc(response=trainY, predictor=LPredProbs)

auc(FullRoc)

**# PLSDA**

set.seed(1)

ctrl <- trainControl(method = "LGOCV",summaryFunction =twoClassSummary,classProbs = TRUE,savePredictions = TRUE)

plsModel <- train(x = trainX,

y = trainY,

method = "pls",

tuneGrid = expand.grid(.ncomp = 1:26),

preProc = c("center","scale"),

metric = "auc",

trControl = ctrl)

plot(plsModel,main="ROC of PLSDA Model")

plsPredProbs<-predict(plsModel, newdata=trainX,type="prob")

plsRoc<-multiclass.roc(response=trainY, predictor=plsPredProbs)

auc(plsRoc)

**#Penalized Model**

ctrl <- trainControl(method = "LGOCV",

summaryFunction = twoClassSummary,

classProbs = TRUE,

##index = list(simulatedTest[,1:4]),

savePredictions = TRUE)

glGrid <- expand.grid(.alpha = c(0, .1, .2, .4, .6, .8, 1),

.lambda = seq(.01, .2, length = 40))

set.seed(476)

glTune<- train(x=trainX,

y = trainY,

method = "glmnet",

tuneGrid = glGrid,

preProc = c("center", "scale"),

metric = "auc",

trControl = ctrl)

glTune

plot(glTune,main="Tuning Parameters of Penalized Model")

#Predict on the training set:

gPred<-predict(glTune, trainX)

postResample(pred=gPred,obs=trainY)

confusionMatrix(glTune$pred$pred,glTune$pred$obs)

gPredProbs<-predict(glTune, newdata=trainX,type="prob")

gRoc<-multiclass.roc(response=trainY, predictor=gPredProbs)

auc(gRoc)

**# SVM**

ctrl <- trainControl(summaryFunction = twoClassSummary,

classProbs = TRUE)

sigma <- sigest(as.matrix(trainX))

svmRGrid <- expand.grid(.sigma = sigma[1],

.C = 2^(seq(-2, 7)))

svmModel <- train(x = trainX,

y = trainY,

method = "svmRadial",

metric = "auc",

preProc = c("center", "scale"),

tuneGrid = svmRGrid,

fit = FALSE,

trControl = ctrl)

svmModel

summary(svmModel)

plot(svmModel, main=" Tuning Parameters of\n Support Vector Machine Model ")

#Predict on the training set:

sPred <- predict(svmModel, newdata = trainX,type="raw")

confusionMatrix(data = sPred,reference =trainY)

sPredProbs<-predict(svmModel, newdata=trainX,type="prob")

sRoc<-multiclass.roc(response=trainY, predictor=sPredProbs)

auc(sRoc) #0.819

#Predict on the test set:

svmPred <- predict(svmModel, newdata = testX,type="raw")

confusionMatrix(data = svmPred,reference =testY)

svmPredProbs<-predict(svmModel, newdata=testX,type="prob")

svmRoc<-multiclass.roc(response=testY, predictor=svmPredProbs)

auc(svmRoc) #0.7151

#Important Variable

varImp(svmModel)

**#Flexible Discriminant Analysis**

marsGrid <- expand.grid(.degree = 1:2, .nprune = 2:30)

fdaTuned <- train(x = trainX,

y = trainY,

method = "fda",

preProcess = c("center", "scale"),

# Explicitly declare the candidate models to test

tuneGrid = marsGrid,

trControl = trainControl(method = "cv"))

fdaTuned

plot(fdaTuned, main="Tuning Parameters of \n Flexible Discriminant Analysis Model ")

#Predict on the training set:

fPred <- predict(fdaTuned, newdata = trainX,type="raw")

confusionMatrix(data = fPred,reference =trainY)

fPredProbs<-predict(fdaTuned, newdata=trainX,type="prob")

fRoc<-multiclass.roc(response=trainY, predictor=fPredProbs)

auc(fRoc)

**#Neural Networks**

library(caret)

Grid <- expand.grid(.size = 1:10, .decay = c(0, .1, .3, .5, 1))

maxSize <- max(Grid$.size)

dim(trainX) #229 26

numWts <- (maxSize \* (26 + 1) + (maxSize+1)\*2) ## 26 is the number of predictors; 2 is the number of classes

set.seed(476)

ctrl <- trainControl(summaryFunction = twoClassSummary,

classProbs = TRUE)

nnetFit <- train(x = trainX,

y = trainY,

method = "nnet",

metric = "auc",

preProc = c("center", "scale", "spatialSign"),

tuneGrid = Grid,

trace = FALSE,

maxit = 2000,

MaxNWts = numWts,

trControl = ctrl)

nnetFit

plot(nnetFit,main="Tuning Parameters of \n Neural Networks Model")

#Predict on the training set:

nPred<-predict(nnetFit, trainX)

confusionMatrix(nPred,trainY)

nPredProbs<-predict(nnetFit , newdata=trainX,type="prob")

nRoc<-multiclass.roc(response=trainY, predictor=nPredProbs)

auc(nRoc)#0.7997

**#KNN**

set.seed(123)

ctrl<-trainControl(summaryFunction=twoClassSummary, classProbs=TRUE)

knnTune <- train(x = trainX,

y = trainY,

method = "knn",

metric = "auc",

preProc = c("center", "scale"),

##tuneGrid = data.frame(.k = c(4\*(0:5)+1, 20\*(1:5)+1, 50\*(2:9)+1)), ## 21 is the best

tuneGrid = data.frame(.k = 1:50),

trControl = ctrl)

knnTune

plot(knnTune,main="Tuning Parameters of KNN Model")

#Predict on the training set:

knnTune$results$Accuracy

kPred<-predict(knnTune, trainX)

postResample(pred=kPred,obs=trainY)

confusionMatrix(kPred,trainY)

kPredProbs<-predict(knnTune, newdata=trainX,type="prob")

kRoc<-multiclass.roc(response=trainY, predictor=kPredProbs)

auc(kRoc) # 0.7581

**# Naive Bayes**

set.seed(123)

nbTune <- train( x = trainX,

y = trainY,

method = "naive\_bayes",

metric = "ROC",

preProc = c("center", "scale"),

#tuneGrid =expand.grid(usekernel=TRUE, adjust=1,fL=c(0.2,0.5,0.8)),

# tuneGrid=data.frame(.fL = 2,.adjust=1,.usekernel = TRUE,.adjust = TRUE),

tuneGrid=data.frame(.laplace =c(0,0.5,1.0,2.0) ,.adjust=c(0,0.5,1.0,2.0),.usekernel = TRUE),

trControl = trainControl(method="cv",number=5,classProbs=TRUE,))

nbTune

plot.new()

plot(nbTune,main="Tuning Parameters of \n Naive Bayes Model")

#Predict on the training set:

nPred<-predict(nbTune, trainX)

postResample(pred=nPred,obs=trainY)

confusionMatrix(nPred,trainY)

nPredProbs<-predict(nbTune, newdata=trainX,type="prob")

nRoc<-multiclass.roc(response=trainY, predictor=nPredProbs)

auc(nRoc) # 0.8285

#Predict on the test set:

nbPred<-predict(nbTune, testX)

postResample(pred=nbPred,obs=testY)

confusionMatrix(nbPred,testY)

nbPredProbs<-predict(nbTune, newdata=testX,type="prob")

nbRoc<-multiclass.roc(response=testY, predictor=nbPredProbs)

auc(nbRoc)

**Resource**

The code of this project is from lectures of Dr. Qiuying Sha and from the below book with some modifications.

Kuhn, Johnson. (2016). Applied Predictive Modeling. New York: Springer

The data was download from:

[Breast Cancer Dataset (kaggle.com)](https://www.kaggle.com/datasets/faysalmiah1721758/breast-cancer-data)