Project01

Pramath Shukla

2024-02-09

```
suppressPackageStartupMessages({
  library(caret)
  library(plotly)
  library(pROC)
  library(ggplot2)
})
```

Dataset

The dataset used for this project is Differentiated Thyroid Cancer Recurrence. There are 16 clinicopathologic features/variables are used to predict recurrence of thyroid cancer. The target variable is classified into two types “yes” or “no” depending on whether there was recurrence of cancer. If there was cancer recurrence, then the output is yes, otherwise no.

Problem

Based on the dataset, how can we predict the recurrence of cancer which is dependent on the set of features designed in the dataset? How well these features and data analysis techniques be used to make such a model and further make it more accurate?

Data set up

```
dataset <- read.csv("Thyroid_Diff.csv")
sum(is.na(dataset))
```

```
[1] 0
```

The above code loads the Thyroid_Diff data into dataset variable. It checks if there are any missing values in the dataset and as the data was already processed, the final dataset doesn’t contains any missing values.

Splitting Data

```
#Let's first visualize the structure of the dataset
str(dataset)
```

```
'data.frame':   383 obs. of  17 variables:
 $ Age                 : int  27 34 30 62 62 52 41 46 51 40 ...
 $ Gender              : chr  "F" "F" "F" "F" ...
 $ Smoking             : chr  "No" "No" "No" "No" ...
 $ Hx.Smoking          : chr  "No" "Yes" "No" "No" ...
 $ Hx.Radiothreapy     : chr  "No" "No" "No" "No" ...
 $ Thyroid.Function    : chr  "Euthyroid" "Euthyroid" "Euthyroid" "Euthyroid" ...
 $ Physical.Examination: chr  "Single nodular goiter-left" "Multinodular goiter" "Single nodular goiter
 $ Adenopathy          : chr  "No" "No" "No" "No" ...
 $ Pathology           : chr  "Micropapillary" "Micropapillary" "Micropapillary" "Micropapillary" ...
 $ Focality            : chr  "Uni-Focal" "Uni-Focal" "Uni-Focal" "Uni-Focal" ...
 $ Risk                : chr  "Low" "Low" "Low" "Low" ...
 $ T                   : chr  "T1a" "T1a" "T1a" "T1a" ...
 $ N                   : chr  "N0" "N0" "N0" "N0" ...
 $ M                   : chr  "M0" "M0" "M0" "M0" ...
 $ Stage               : chr  "I" "I" "I" "I" ...
 $ Response            : chr  "Indeterminate" "Excellent" "Excellent" "Excellent" ...
 $ Recurred            : chr  "No" "No" "No" "No" ...
```

```r
set.seed(123)

dataset$Age <- as.integer(dataset$Age)
dataset$Gender <- as.factor(dataset$Gender)
dataset$Smoking <- as.factor(dataset$Smoking)
dataset$Hx.Smoking <- as.factor(dataset$Hx.Smoking)
dataset$Hx.Radiothreapy <- as.factor(dataset$Hx.Radiothreapy)
dataset$Thyroid.Function <- as.factor(dataset$Thyroid.Function)
dataset$Physical.Examination <- as.factor(dataset$Physical.Examination)
dataset$Adenopathy <- as.factor(dataset$Adenopathy)
dataset$Pathology <- as.factor(dataset$Pathology)
dataset$Focality <- as.factor(dataset$Focality)
dataset$Risk <- as.factor(dataset$Risk)
dataset$T <- as.factor(dataset$T)
dataset$N <- as.factor(dataset$N)
dataset$M <- as.factor(dataset$M)
dataset$Stage <- as.factor(dataset$Stage)
dataset$Response <- as.factor(dataset$Response)
dataset$Recurred <- as.factor(dataset$Recurred)

spec = c(trainData = .6, testData = .2, cvData = .2)

g = sample(cut(
  seq(nrow(dataset)),
  nrow(dataset)*cumsum(c(0,spec)),
  labels = names(spec)
))

res = split(dataset, g)
trainData <- res$trainData
testData <- res$testData
cvData <- res$cvData
```

First, we can see the structure of the data set, with 17 variables, out of which 16 are independent variables and "Recurred" is the output or the target variable. Then I have sorted the variables as numerical and categorical. Moreover, the categorical variables are factored accordingly with different levels as per the number of categories present in the variable. The data is splitted into three sets which are training, cross validation and testing sets. The training data is 60%, cross-validation data is 20% and the remaining 20% is testing data. The data will be first trained and then will be cross-validated to evaluate the model and to make adjustments such that it does not overfits and neither underfits. Moreover, only after building proper model, it is then tested on the testing set.

Checking the data consistency

```
# check if both the genders have cancer recurrence
xtabs(~ Recurred + Gender, data = dataset)
```

```
        Gender
Recurred   F   M
     No  246  29
     Yes  66  42
```

```
# check for other variables with respect to Recurrence
xtabs(~ Recurred + Hx.Radiothreapy, data = dataset)
```

```
        Hx.Radiothreapy
Recurred  No Yes
     No  274   1
     Yes 102   6
```

```
xtabs(~ Recurred + Thyroid.Function, data = dataset)
```

```
        Thyroid.Function
Recurred Clinical Hyperthyroidism Clinical Hypothyroidism Euthyroid
     No                        17                      10       234
     Yes                        3                       2        98
        Thyroid.Function
Recurred Subclinical Hyperthyroidism Subclinical Hypothyroidism
     No                            5                          9
     Yes                           0                          5
```

```
xtabs(~ Recurred + Physical.Examination, data = dataset)
```

```
        Physical.Examination
Recurred Diffuse goiter Multinodular goiter Normal Single nodular goiter-left
     No               7                  88      5                         63
     Yes              0                  52      2                         26
        Physical.Examination
Recurred Single nodular goiter-right
     No                          112
     Yes                          28
```

```
xtabs(~ Recurred + Adenopathy, data = dataset)
```

```
        Adenopathy
Recurred Bilateral Extensive Left  No Posterior Right
     No          5         0    5 247         0    18
     Yes        27         7   12  30         2    30
```

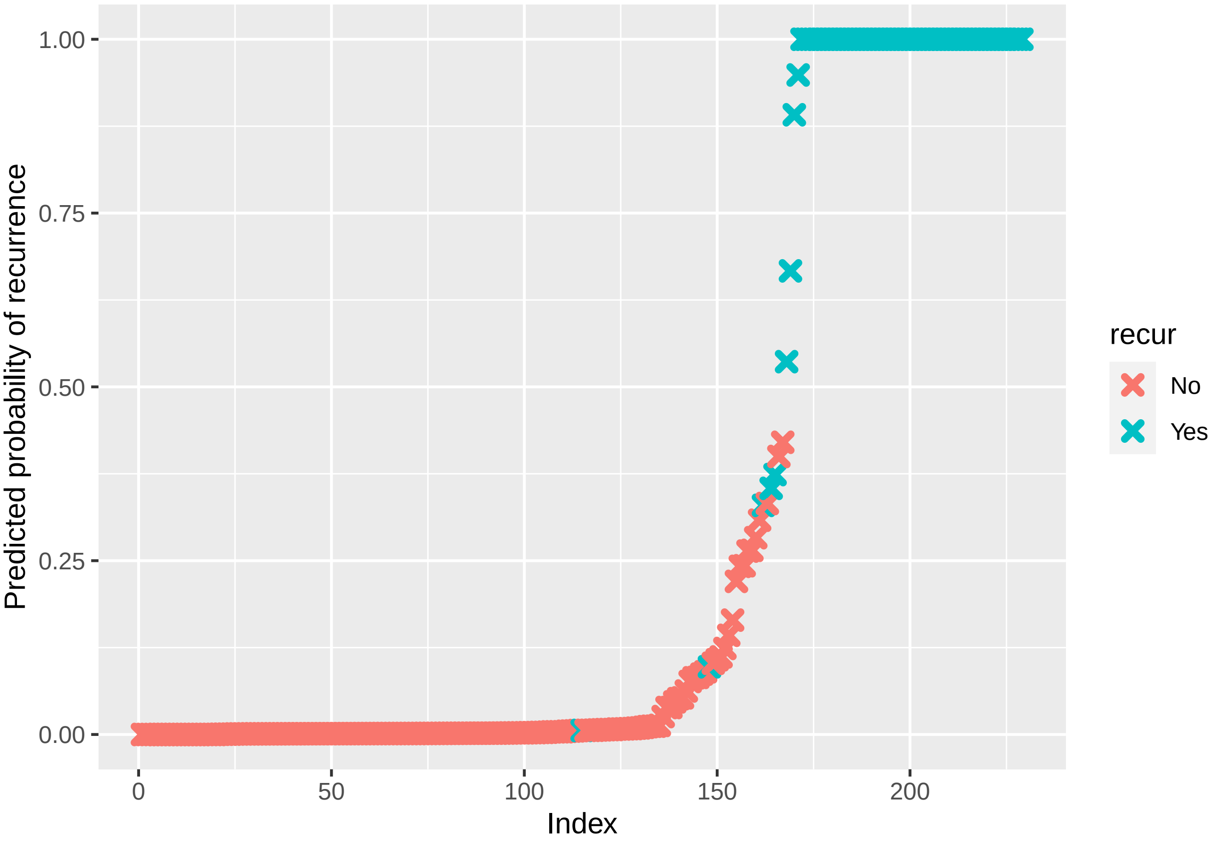
```
xtabs(~ Recurred + Stage, data = dataset)
```

```
        Stage
Recurred   I  II III IVA IVB
     No  268   7   0   0   0
     Yes  65  25   4   3  11
```

Here, I have checked if the data of recurrence is consistent across different parameters such as Gender, Thyroid.Function and so on. Moreover, except Gender, I have only presented the data which seemed to be quite inconsistent like Hx.Radiothreapy parameter has a lot of inconsistency. It seems like that parameter is not of much importance because it makes little difference to the Recurrence. However, we will look more closely into many of the variables by visualizing them later.

Building and plotting the training model

```
# fitting the logistic regression model
logistic_model <- glm(Recurred ~ Gender+Age+Smoking+Thyroid.Function+Focality+N+Response+Stage+Hx.Radiot

predicted.data <- data.frame(
  probability.of.Recur=logistic_model$fitted.values,
  recur = trainData$Recurred
)

predicted.data <- predicted.data[
  order(predicted.data$probability.of.Recur, decreasing=FALSE),]
predicted.data$rank <- 1:nrow(predicted.data)

ggplot(data=predicted.data, aes(x=rank, y=probability.of.Recur))+
  geom_point(aes(color=recur), alpha=1, shape=4, stroke=2)+
  xlab("Index")+
  ylab("Predicted probability of recurrence")
```



```
cv_prediction <- predict(logistic_model, cvData, type = "response")
cv_prediction <- ifelse(cv_prediction> 0.5,1,0)
```

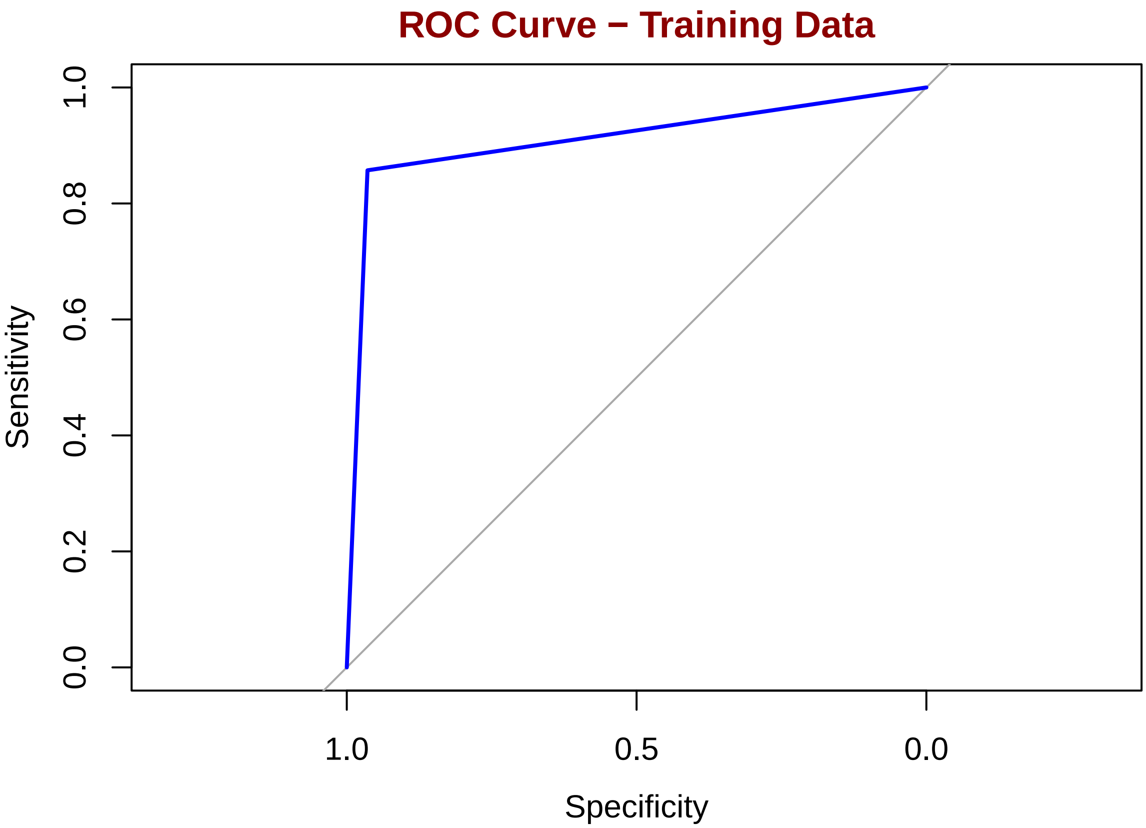
Here, I have built a logistic model with glm function with almost all the parameters. Then the model is plotted using ggplot to see how well it fits the training data. The observations of the data has been sequentially ordered with lower probabilities having lower rank and thus, lower chances of getting recurrence while, the higher probabilities have higher rank and thus, higher chances of getting cancer recurrence. It seems that model is doing well so far. It forms a good logistic plot, where it has managed to get most of the cases correctly. Thus, we can go ahead with checking how well it does on the cross-validation data set using roc plot.

```
# Create ROC curve
roc_cv <- roc(cvData$Recurred, cv_prediction)
```

```
Setting levels: control = No, case = Yes
```

```
Setting direction: controls < cases
```

```
# Plot ROC curve for training data
plot(roc_cv, col = "blue", main = "ROC Curve - Training Data", col.main = "darkred", lwd = 2)
```



```
# Calculate the AUC
auc_value <- auc(roc_cv)
cat("AUC:", auc_value, "\n")
```

```
AUC: 0.9107143
```

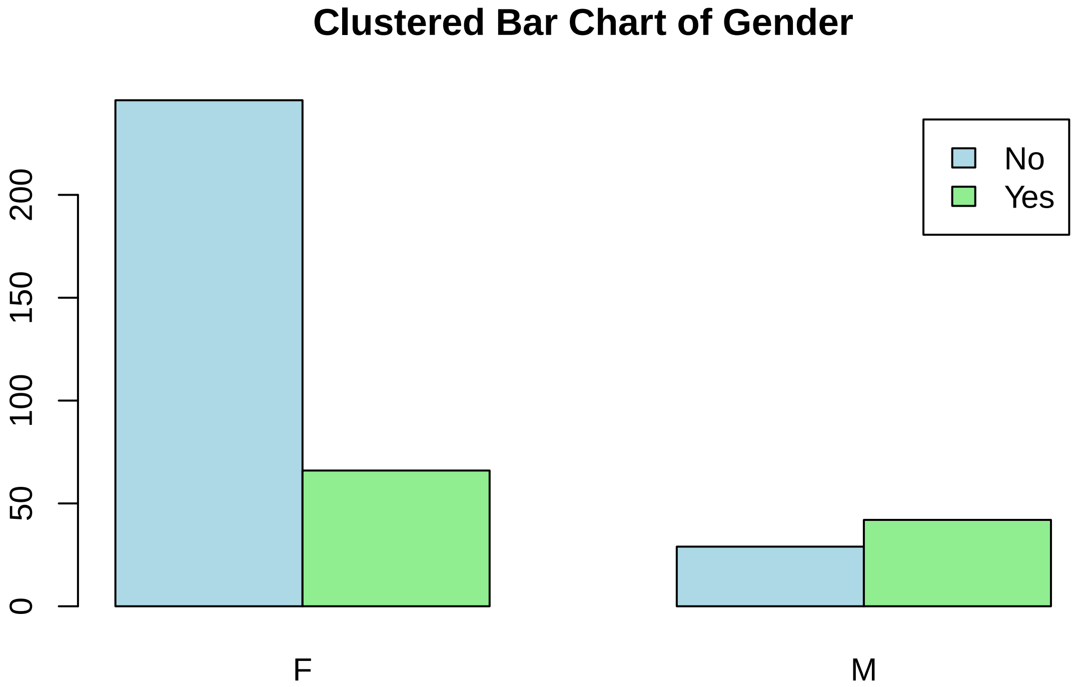
The above roc plot tells us how well our model is doing with the training data. Well, it turns it did great! The plot is well towards the upper-left corner with high Sensitivity(True positive rates) and low Specificity. Moreover, we can further evaluate the model performace through AUC, which is the area under the ROC curve and summarizes the performance of the classifier. Where, the AUC value of the model is 0.9107! (On scale of, where 1.0 is a perfect model). These are pretty great statistics, we now see how we can further enhance the prediction of the model through data wrangling and visualization.

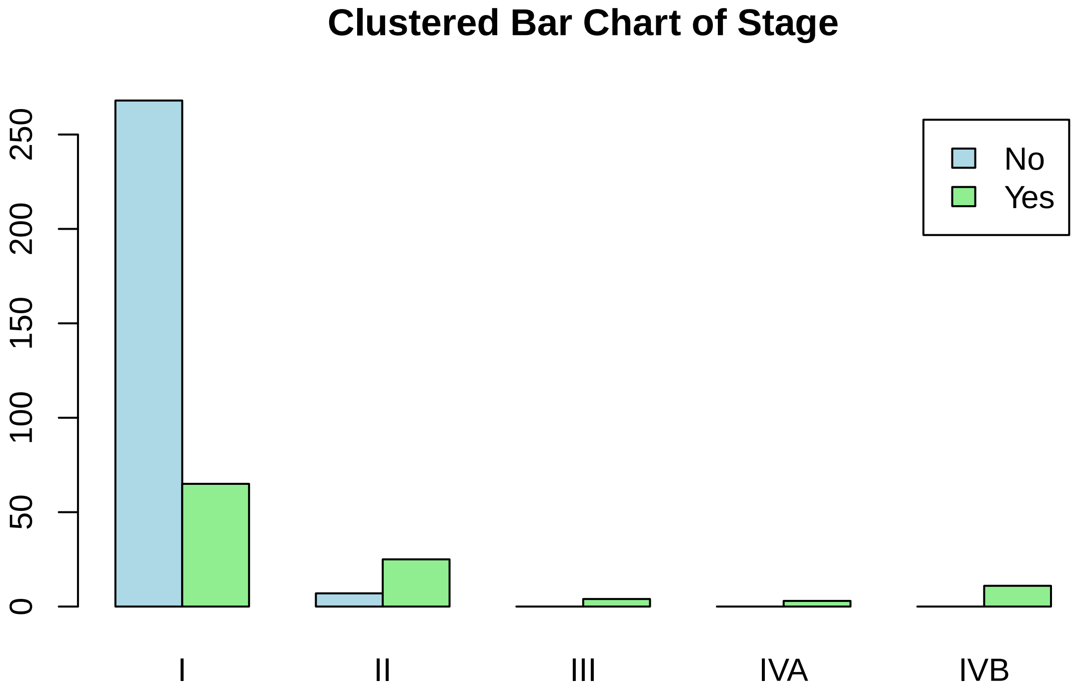
Evaluating the model

```r
# Model summary
summary(logistic_model)
```

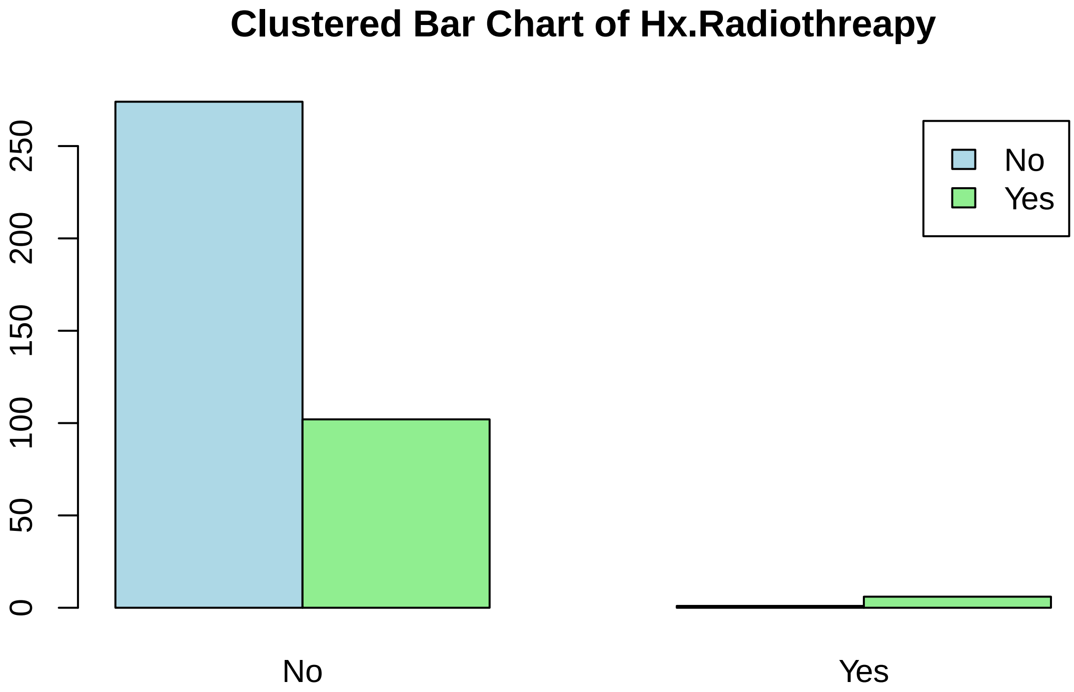
```

Call:
glm(formula = Recurred ~ Gender + Age + Smoking + Thyroid.Function + 
    Focality + N + Response + Stage + Hx.Radiothreapy, family = "binomial", 
    data = trainData, maxit = 1000)

Coefficients:
                                              Estimate Std. Error z value
(Intercept)                                 -5.341e+01  1.300e+04  -0.004
GenderM                                      5.414e-01  1.336e+00   0.405
Age                                          3.044e-02  4.376e-02   0.696
SmokingYes                                  -1.544e+00  3.471e+00  -0.445
Thyroid.FunctionClinical Hypothyroidism      5.174e+01  1.300e+04   0.004
Thyroid.FunctionEuthyroid                    5.008e+01  1.300e+04   0.004
Thyroid.FunctionSubclinical Hyperthyroidism  3.409e+01  2.489e+04   0.001
Thyroid.FunctionSubclinical Hypothyroidism   3.432e+01  1.241e+04   0.003
FocalityUni-Focal                            4.800e-01  1.041e+00   0.461
NN1a                                         1.957e+00  1.669e+00   1.173
NN1b                                         4.427e+00  1.561e+00   2.836
ResponseExcellent                           -4.751e+00  1.632e+00  -2.911
ResponseIndeterminate                       -2.906e+00  1.489e+00  -1.952
ResponseStructural Incomplete                3.025e+01  5.534e+03   0.005
StageII                                      3.320e+00  2.019e+00   1.644
StageIII                                     4.437e+00  2.441e+04   0.000
StageIVA                                     3.986e+01  2.543e+04   0.002
StageIVB                                     3.084e+01  2.136e+04   0.001
Hx.RadiothreapyYes                          -1.359e+01  1.655e+04  -0.001
                                            Pr(>|z|)   
(Intercept)                                  0.99672   
GenderM                                      0.68522   
Age                                          0.48669   
SmokingYes                                   0.65648   
Thyroid.FunctionClinical Hypothyroidism      0.99682   
Thyroid.FunctionEuthyroid                    0.99693   
Thyroid.FunctionSubclinical Hyperthyroidism  0.99891   
Thyroid.FunctionSubclinical Hypothyroidism   0.99779   
FocalityUni-Focal                            0.64473   
NN1a                                         0.24096   
NN1b                                         0.00456 **
ResponseExcellent                            0.00360 **
ResponseIndeterminate                        0.05095 . 
ResponseStructural Incomplete                0.99564   
StageII                                      0.10013   
StageIII                                     0.99985   
StageIVA                                     0.99875   
StageIVB                                     0.99885   
Hx.RadiothreapyYes                           0.99934   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 276.83  on 228  degrees of freedom
Residual deviance:  34.56  on 210  degrees of freedom
AIC: 72.56

Number of Fisher Scoring iterations: 21
```

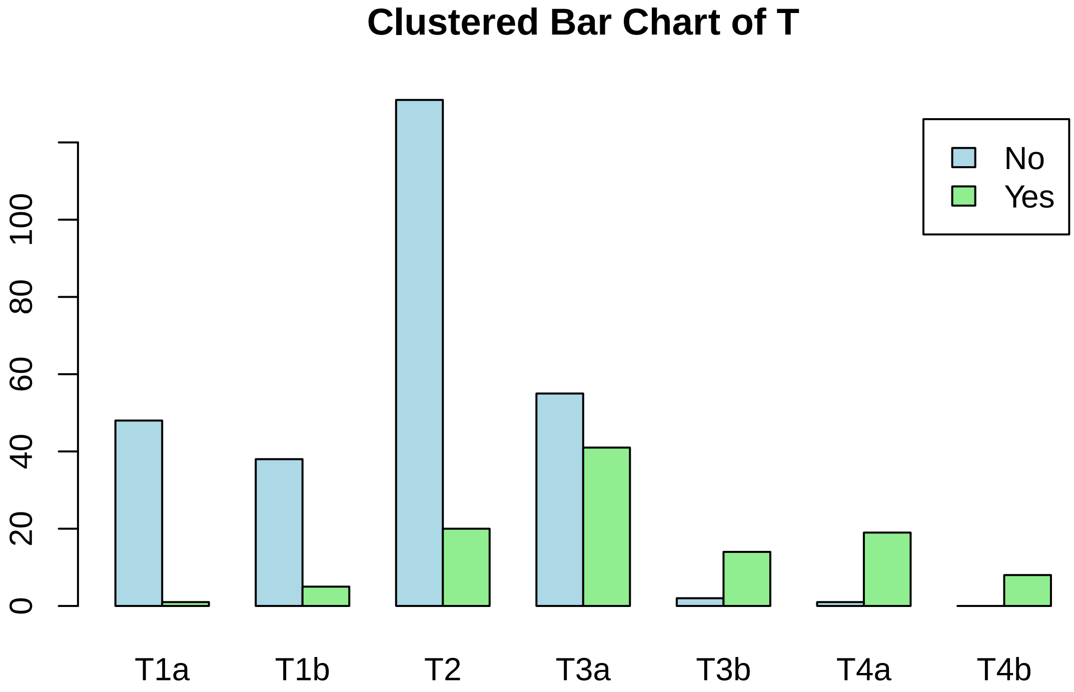




The above bar chart displays the association between the “Stage” variable and the “Recurred” output variable. The variable doesn’t provides a strong relation with respect to recurrence. Especially, in the first stage, most of the cases have no recurrence and it has little co-relation in higher stages. Therefore, it seems to be better to get rid of the variable from the model.



The Hx.Radiothreapy is very inconsistent. It is because, there are VERY less cases in which the threapy was done as it is apparent from the plot. Moreover, including this variable can lead to inconsistencies when applied on unseen data.



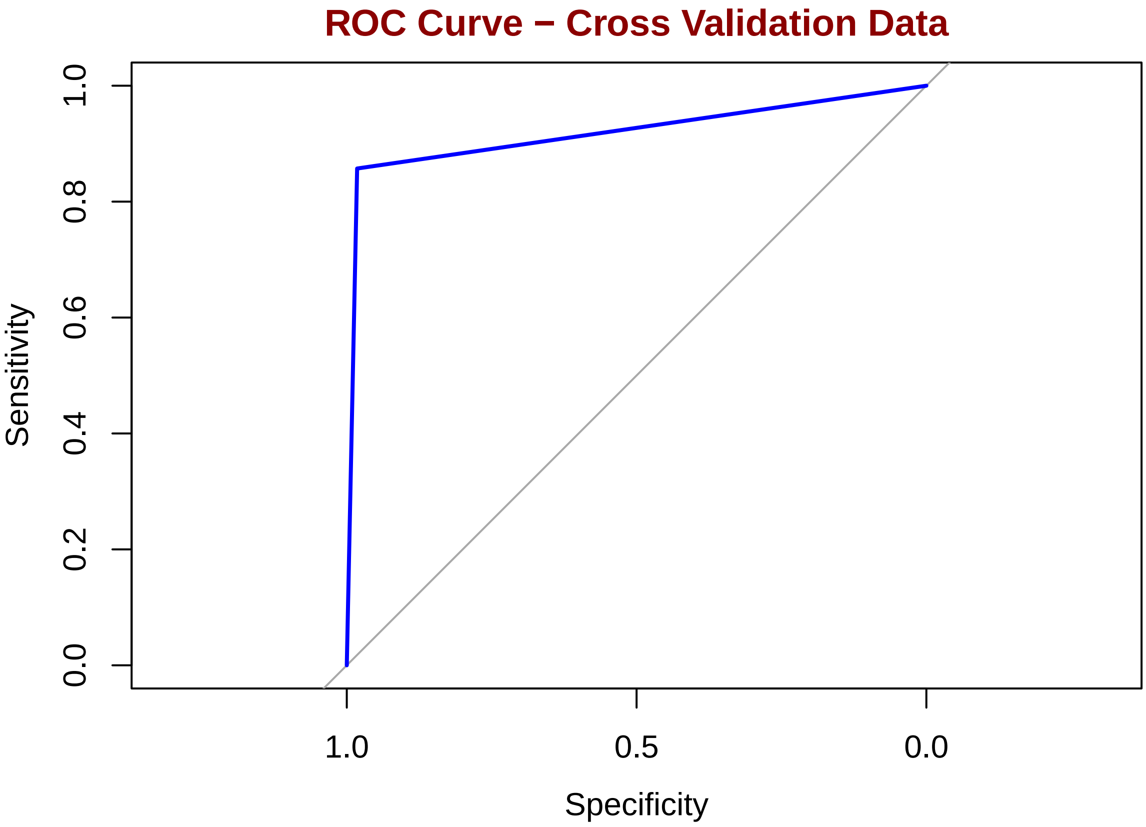
The T variable that was not included earlier, seems to be giving good co-relation with the output variable of Recurred. Therefore, it would be a better option to include the variable during model training.

Improving the Model

```
# fitting the logistic regression when considering all the predictors
model <- glm(Recurred ~ Gender+Age+Smoking+Thyroid.Function+Focality+N+T+Response, trainData, family =

cv_prediction1 <- predict(model, cvData, type = "response")
cv_prediction1 <- ifelse(cv_prediction1> 0.5,1,0)

# Create ROC curve
roc_cv1 <- roc(cvData$Recurred, cv_prediction1)
```

```
Setting levels: control = No, case = Yes

Setting direction: controls < cases
```

```
# Plot ROC curve for training data
plot(roc_cv1, col = "blue", main = "ROC Curve - Cross Validation Data", col.main = "darkred", lwd = 2)
```

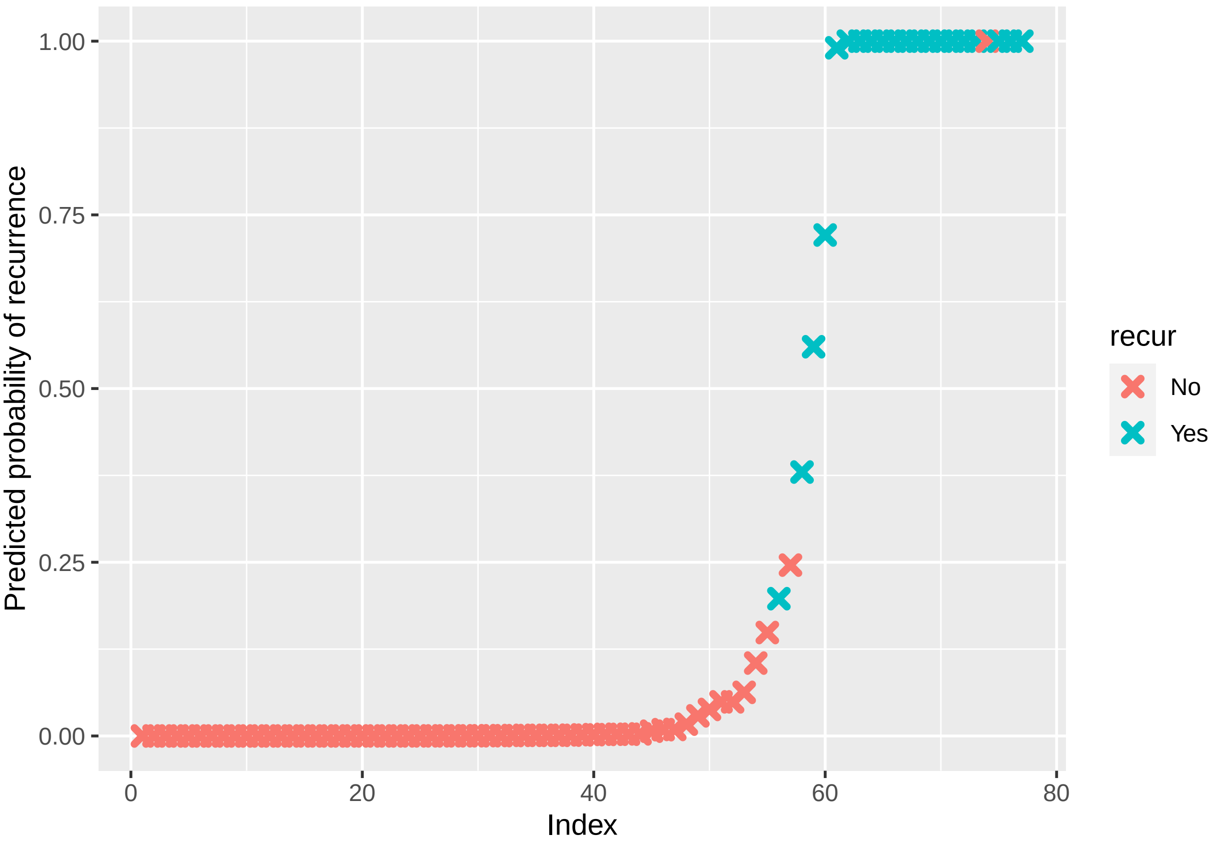


```
# Calculate the AUC
auc_value1 <- auc(roc_cv1)
cat("AUC:", auc_value1, "\n")
```

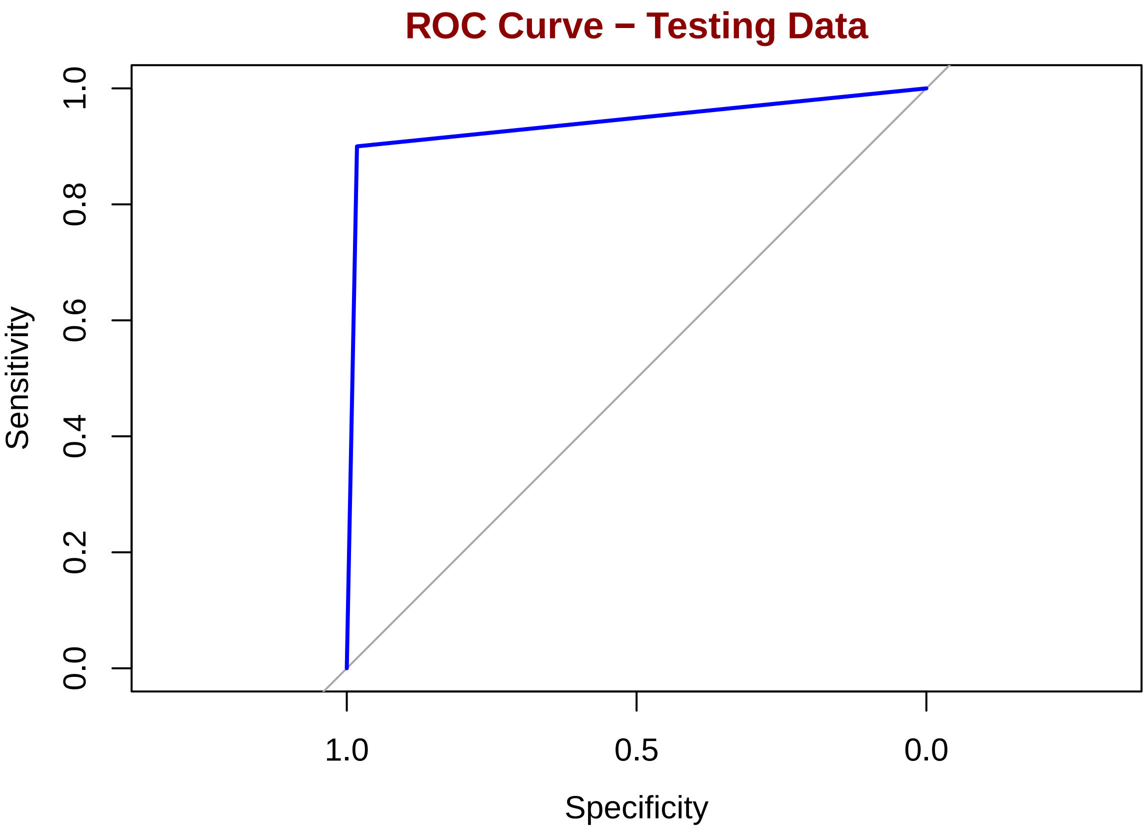
```
AUC: 0.9196429
```

The performance of the model has improved since the last time. The roc value increased from 0.9107 to 0.9192. Thus, it signals that we are ready to test it on our final test dataset. Let's see how it performs.

Testing the model



```
Setting levels: control = No, case = Yes

Setting direction: controls < cases
```



```
AUC: 0.9412281
```

0.9412! That's seem to be great with the data that it has not seen before. Moreover, the output column was removed from the testing data so that the model has no means to know the values of the target data in the testing data set.

Conclusion

With rigorous model training using training and cross validation data, as well as data wrangling and visualization techniques, we were able to come up with model that was able to estimate cancer recurrence in patients with high probability and precision.

Data Citation

JOAKIM ARVIDSSON. ([2024; 01]). "Differentiated Thyroid Cancer Recurrence", Version 1. Retrieved 02/05/2024 from https://www.kaggle.com/datasets/joebeachcapital/differentiated-thyroid-cancer-recurrence.