Predicting Breast Cancer using logistic regression classification

Code

Hide

```
library(pROC)
```

```
Type 'citation("pROC")' for a citation.

Attaching package: <U+393C><U+3E31>pROC<U+393C><U+3E32>

The following object is masked from <U+393C><U+3E31>package:h2o<U+393C><U+3E32>:

    var

The following objects are masked from <U+393C><U+3E31>package:stats<U+393C><U+3E32>:

    cov, smooth, var
```

Hide

```
# Importing the dataset
dataset = read.csv('breast_cancer_prediction.csv')
dataset = data.frame(dataset)
# Encoding the target feature as factor
dataset$Classification = ifelse(dataset$Classification==1, 0,1)
head(dataset)
```

	...	BMI	Glucose	Insulin	HOMA	Leptin	Adiponectin	Resistin	MCP.1
	<int>	<dbl>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	48	23.50000	70	2.707	0.4674087	8.8071	9.702400	7.99585	417.114
2	83	20.69049	92	3.115	0.7068973	8.8438	5.429285	4.06405	468.786
3	82	23.12467	91	4.498	1.0096511	17.9393	22.432040	9.27715	554.697
4	68	21.36752	77	3.226	0.6127249	9.8827	7.169560	12.76600	928.220
5	86	21.11111	92	3.549	0.8053864	6.6994	4.819240	10.57635	773.920
6	49	22.85446	92	3.226	0.7320869	6.8317	13.679750	10.31760	530.410

6 rows | 1-10 of 10 columns

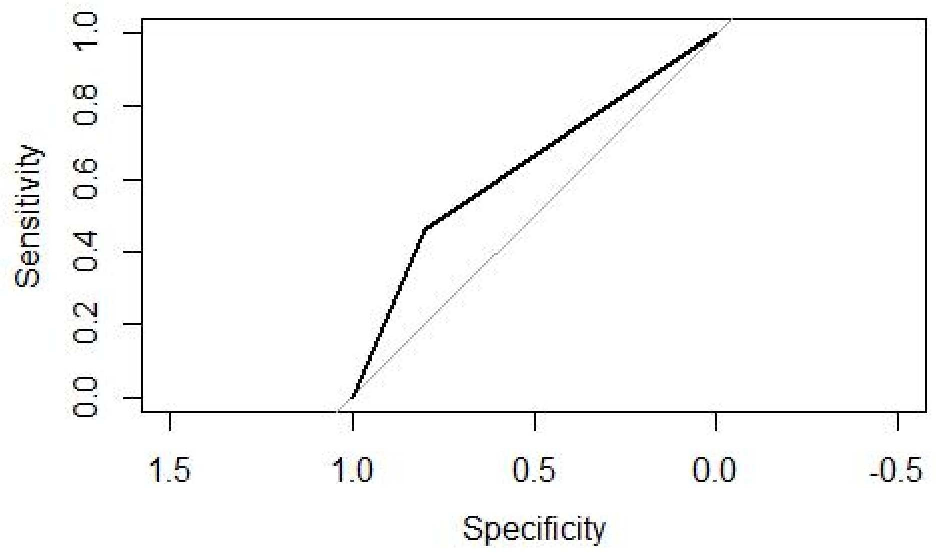
Hide

```
# Splitting the dataset into the Training set and Test set
# install.packages('caTools')
library(caTools)
set.seed(123)
split = sample.split(dataset$Classification, SplitRatio = 0.8)
training_set = subset(dataset, split == TRUE)
test_set = subset(dataset, split == FALSE)
# Feature Scaling
out_index = which(colnames(dataset)=="Classification")
training_set[-out_index] = scale(training_set[-out_index])
test_set[-out_index] = scale(test_set[-out_index])
# Fitting Logistic Regression to the Training set
classifier = glm(formula = Classification ~ .,
                 family = binomial,
                 data = training_set)
summary(classifier)
```

```

Call:
glm(formula = Classification ~ ., family = binomial, data = training_set)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.1095  -0.7409   0.1632   0.6544   2.2083  

Coefficients:
            Estimate Std. Error z value Pr(>|z|)  
(Intercept)   0.5019     0.3670   1.368   0.1714  
Age          -0.3806     0.2888  -1.318   0.1876  
BMI          -0.6685     0.3778  -1.769   0.0768 .
Glucose       2.0568     0.8016   2.566   0.0103 *
Insulin       1.9181     3.1383   0.611   0.5411  
HOMA         -1.5071     4.2230  -0.357   0.7212  
Leptin       -0.4548     0.3811  -1.193   0.2327  
Adiponectin   0.1215     0.3107   0.391   0.6958  
Resistin      0.8091     0.3739   2.164   0.0305 *
MCP.1         0.2876     0.3165   0.909   0.3635  
---
Signif. codes:  0 ┘***└ 0.001 └**└ 0.01 ┘*└ 0.05 _.┘ 0.1 ┘ ┘ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 128.053  on 92  degrees of freedom
Residual deviance:  82.904  on 83  degrees of freedom
AIC: 102.9

Number of Fisher Scoring iterations: 7
```

Hide

```
# Interesting results from the model, blood glucose levels and resistin levels are statistically
 significant at p<0.05 in predicting breast cancer in trainign data.
# Predicting the Test set results
prob_pred = predict(classifier, type = 'response', newdata = test_set[-out_index])
y_pred = ifelse(prob_pred > 0.5, 1, 0)
# ROC Curve
library(pROC)
preds=predict(classifier,test_set[-out_index], type="response")
ro <- roc(test_set[,out_index] ~ y_pred)
plot(ro)
```



```
auc(ro)
```

```
Area under the curve: 0.6308
```

```
#Model has 63% accuracy o test data, seems to have better predictive power than random choice.
# Evaluating Model Accuracy on test data set using
# Confusion Matrix
cm = table(test_set[, out_index], y_pred > 0.5)
print(cm)
```

```
    FALSE TRUE
  0     8    2
  1     7    6
```

```
Model_Accuracy=(cm[1,1]+cm[2,2])/(cm[1,1]+cm[1,2]+cm[2,1]+cm[2,2])
print("Assuming 50% probability as cutoff,  Model Accuracy is")
```

```
[1] "Assuming 50% probability as cutoff,  Model Accuracy is"
```

Hide

```
print(Model_Accuracy)
```

```
[1] 0.6086957
```