Logistic Regression

Links: MATH 4322

Info

Logistic Regression can be used to model and solve problems when the response variable, *Y*, is a categorical with *2 classes*.
("Introduction to classification is how you can think about this" ~ Prof. Poliak).

This is also called binary classification problems. This models the **probability** that Y belongs to one of the two categories.

We're trying to predict in which class is the output, in regression we try to predict the mean response based on the *x* variables. In classification we can't get a mean, but we can get a probability of being in a particular class.

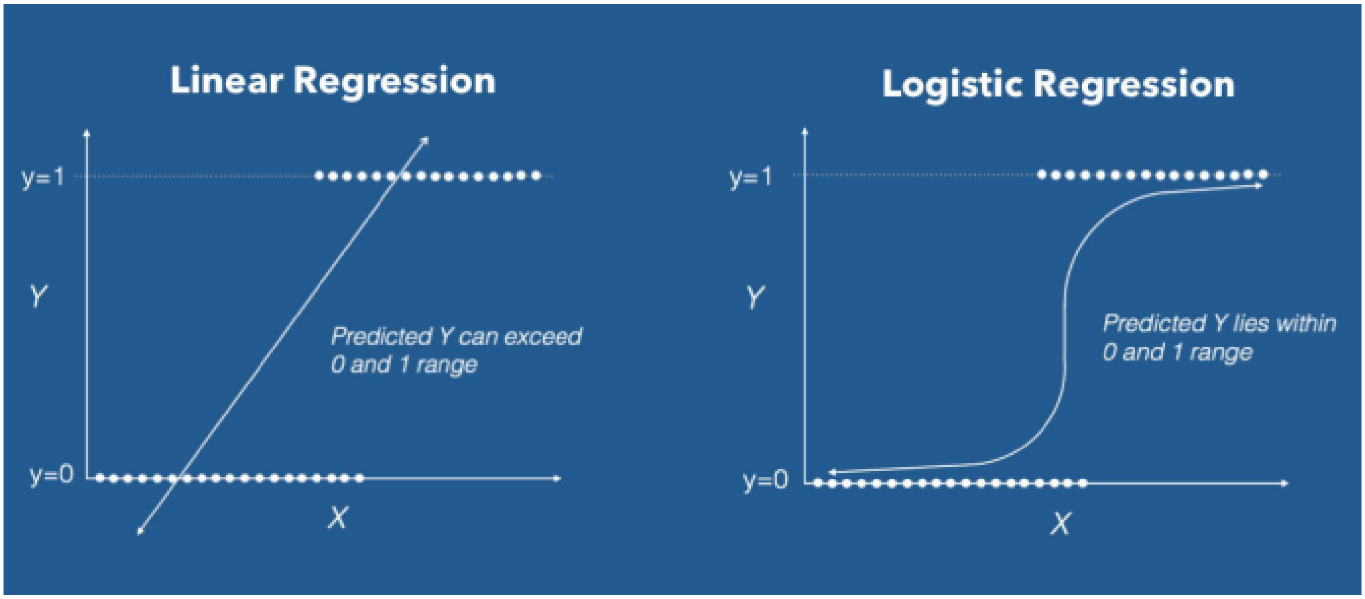
"The classification problem has to be that your output is categorical, not necessarily that all your values are categorical, just your output must be categorical" ~ Prof. Poliak.

examples of binary classification problems:

- Spam Detection : Predicting if an email is Spam or not
- Credit Card Fraud : Predicting if a given credit card transaction is fraud or not
- Health : Predicting if a given mass of tissue is benign or malignant
- Marketing : Predicting if a given user will buy an insurance product or not
- Banking : Predicting if a customer will default on a loan.

Why not use linear regression?

- When the response variable has only 2 possible values, it is desirable to have a model that predicts the value either as 0 or 1, or as a probability score that ranges between 0 and 1.
- If the linear regression is used to predict a binary response, the resulting model may not restrict the predicted Y values within 0 and 1.


Linear Regression
y=1
Y
y=0
X
Predicted Y can exceed 0 and 1 range
Logistic Regression
y=1
Y
y=0
X
Predicted Y lies within 0 and 1 range

The Logistic Model

Given $Y = 0 \text{ or } 1$, let $p(X) = P(Y = 1|X)$. We want a model that shows the relationship between $p(X) \text{ and } X$. We are trying to predict that the response is equal to 1 given that we know these predictors (recall Conditional Probability).

We use a model that gives outputs between 0 and 1 for all values of *X*. This is called the **logistic function**

$$p(X) = \frac{\exp^{\beta_0 + \beta_1 X}}{1 + \exp^{\beta_0 + \beta_1 X}}$$

After some manipulation we get

$$\frac{p(X)}{1-p(X)} = \exp^{\beta_0+\beta_1 X}$$

How to get that above manipulation

Reciprocate both sides:

$$(p(X))^{-1} = (\frac{\exp^{\beta_0+\beta_1 X}}{1+\exp^{\beta_0+\beta_1 X}})^{-1}$$

$$\frac{1}{p(X)} = \frac{1+\exp^{\beta_0+\beta_1 X}}{\exp^{\beta_0+\beta_1 X}}$$

Split the right hand side and simplify:

$$\frac{1}{p(X)} = \frac{1}{\exp^{\beta_0+\beta_1 X}} + \frac{\exp^{\beta_0+\beta_1 X}}{\exp^{\beta_0+\beta_1 X}}$$

$$\frac{1}{p(X)} = \frac{1}{\exp^{\beta_0+\beta_1 X}} + 1$$

Subtract 1 from both sides:

$$\frac{1}{p(X)} - 1 = \frac{1}{\exp^{\beta_0+\beta_1 X}}$$

-1 is the same as $-\frac{p(X)}{p(X)}$:

$$\frac{1}{p(X)} - \frac{p(X)}{p(X)} = \frac{1}{\exp^{\beta_0+\beta_1 X}}$$

Combine the left hand side:

$$\frac{1-p(X)}{p(X)} = \frac{1}{\exp^{\beta_0+\beta_1 X}}$$

Reciprocate both sides:

$$(\frac{1-p(X)}{p(X)})^{-1} = (\frac{1}{\exp^{\beta_0+\beta_1 X}})^{-1}$$

$$\frac{p(X)}{1-p(X)} = \exp^{\beta_0+\beta_1 X}$$

We can then take the logarithm (natural log) of both sides:

$$\log(\frac{p(X)}{1-p(X)}) = \beta_0 + \beta_1 X$$

The left-hand side is called the *log-odds* or *logit*.

We use a method called the **maximum likelihood** to determine the best coefficients and eventually a good fit.

Maximum Likelihood Method

This method tries to find the value of coefficients (β_0, β_1) such that predicted probabilities are as close to the observed probabilities as possible.

$P(Y=1 \mid x) = p(x)$ for $y_i = 1$ we try to estimate β_0 and β_1 such that $p(x)$ close to 1.

for $y_i = 0$ we try to estimate β_0 and β_1 such that $1-p(x)$ is close to 1.

$p(x)^y [1-p(x)]^{1-y}$

In order words, for a binary classification (1/0), maximum likelihood will try to find values of $\beta_0 \text{ and } \beta_1$ such that the resultant probabilities are closest to either 1 or 0.

The likelihood function is written as

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'}))$$

("We're solving for the product of that Bernoulli random variable" ~ Prof. Poliak)
The estimates $\hat{\beta}_0$ and $\hat{\beta}_1$ are chosen to maximize this likelihood function.

In R we use the function: `glm(model, family="binomial")` function.

Generalized Linear Models

Generalized Linear Models (glm) are an extension of the linear model framework, which includes dependent variables which are non-normal also. In general, they possess three characteristics:

- These models comprise a linear combination of input features.
- The mean of the response variable is related to the linear combination of input features via a link function.
- The response variable is considered to have an underlying probability distribution belonging to the family of exponential distributions such as binomial distribution, Poisson distribution, or Gaussian (aka Normal) distribution.
  - Practically, binomial distribution is used when the response variable is binary.
  - Poisson distribution is used when the response variable represents count.
  - Gaussian distribution is used when the response variable is continuous.

Logistic Regression assumes that the dependent (or response) variable follows a binomial distribution.

Breast Cancer Data Example

from slide (pdf page) 19 of lecture 9 slides

> Using the BreastCancer data set in mlbench package. You will have to install the "mlbench" package for this.
> The response, Y is the **Class** this has two categories malignant and benign.
> We want to use Cell.shape as the predictor.

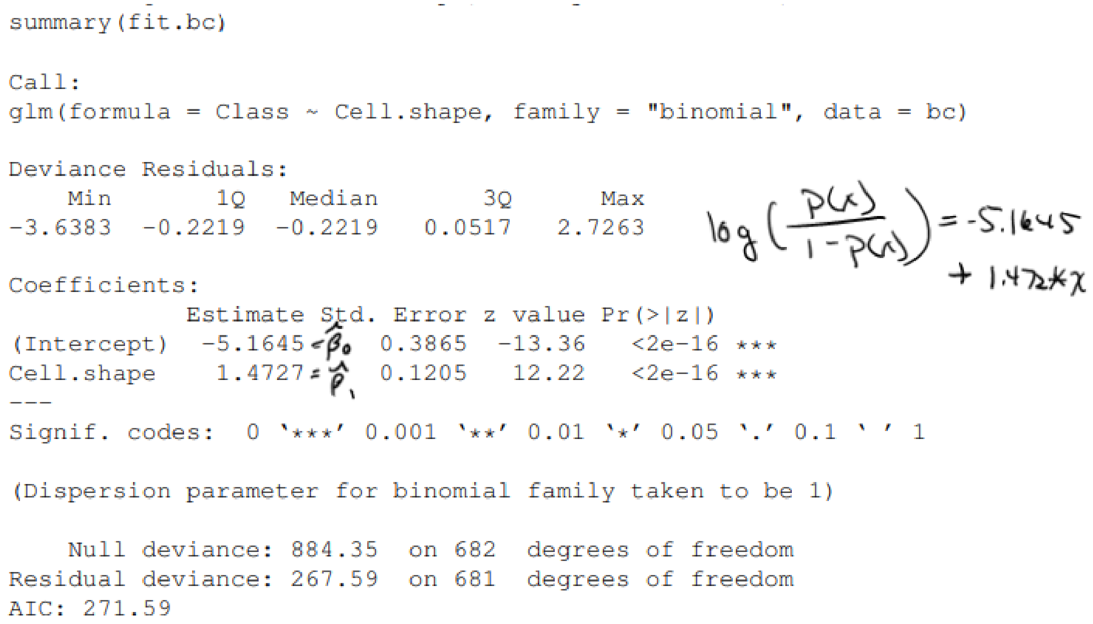
In order to use this data we have to *clean* this data to use with the `glm` function.
Cleaning this data:

```
# no missing rows
bc <- BreastCancer[complete.cases(BreastCancer), ]
bc <- bc[,-1] # remove id column

# convert the factors to numeric
for(i in 1:9) {
   bc[, i] <- as.numeric(as.character(bc[, i]))
}

# put malignant = 1 and benign = 0 (not necessary)
bc$Class = ifelse(bc$Class == "malignant", 1, 0)
bc$Class = as.factor(bc$Class)
```

Create and see the summary of the model with this:

```
fit.bc = glm(Class ~ Cell.shape, family = "binomial", data = bc)
#              ^         ^                    ^
#            y var     x var                logistic reg.
# if we don't do "family = "binomial"" it only does
# regular regression

summary(fit.bc)
```

summary output:



```
summary(fit.bc)

Call:
glm(formula = Class ~ Cell.shape, family = "binomial", data = bc)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6383  -0.2219  -0.2219   0.0517   2.7263

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -5.1645 <β̂0  0.3865  -13.36   <2e-16 ***
Cell.shape    1.4727 = β̂1  0.1205   12.22   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 884.35  on 682  degrees of freedom
Residual deviance: 267.59  on 681  degrees of freedom
AIC: 271.59
```

$$\log\left(\frac{p(x)}{1-p(x)}\right) = -5.1645 + 1.472 * x$$

(Notice we get a *z-value* instead of a *t-value* for the significance of the predictors. This is because we are testing proportions and not means).

our model is

$$\hat{p}(X) = \frac{\exp^{\hat{\beta}_0+\hat{\beta}_1 x}}{1+\exp^{\hat{\beta}_0+\hat{\beta}_1 x}} = \frac{\exp^{-5.1645+1.4727x}}{1+\exp^{-5.1645+1.4727x}}$$

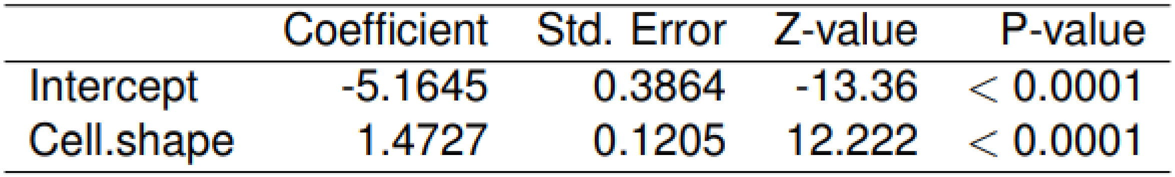
Interpreting the Predictor

Suppose we have a cell shape of 5 (i.e. $P(Y=1|\text{Cell.shape}=5)=?$). This is asking "What is the predicted probability of being malignant given the uniformity of the cell shape (Cell.shape) is 5?"

We could plug in 5 into the above equation and get a result (0.9001597), but we could also calculate this in R:

```
predict.glm(fit.bc, newdata = data.frame(Cell.shape=5),
                        type = "response")

# output:
# 1
# 0.9001648
# if you don't put the 'type = "Response"' you will get the
# log-odds instead
```

Interpreting the Estimated Parameters



	Coefficient	Std. Error	Z-value	P-value
Intercept	-5.1645	0.3864	-13.36	< 0.0001
Cell.shape	1.4727	0.1205	12.222	< 0.0001

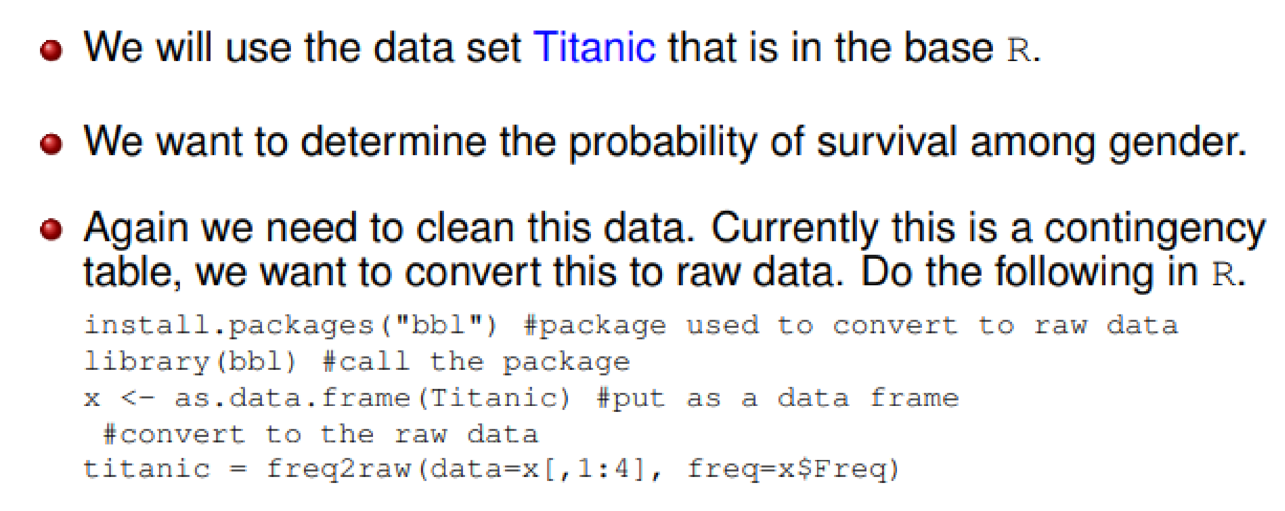
("If the p value was high then we don't need that variable in the model" ~ Prof. Poliak)

- $\hat{\beta}_1 = 1.4727$ indicates that an increase in Cell.shape is associated with an increase of the probability of Class.
- A one unit increase in Cell.shape is associated with an increase in the log odds of Class by 1.4727 units.
- For testing $H_0 : \beta_1 = 0$, this null hypothesis implies that $p(X) = \frac{\exp^{\beta_0}}{1+\exp^{\beta_0}}$. Thus if we fail to reject H_0 the probability of **Class** does not depend on Cell.shape.
- Since p-value for this test is < 0.0001 we reject the null hypothesis and conclude that there is an association between Cell.shape and probability of Class.

Categorical Predictors

The examples in this section will use the Titanic dataset in R, we want to determine the probability of survival among gender.

ⓘ **Cleaning the Data** ›



- We will use the data set Titanic that is in the base R.
- We want to determine the probability of survival among gender.
- Again we need to clean this data. Currently this is a contingency table, we want to convert this to raw data. Do the following in R.

```
install.packages("bbl") #package used to convert to raw data
library(bbl) #call the package
x <- as.data.frame(Titanic) #put as a data frame
 #convert to the raw data
titanic = freq2raw(data=x[,1:4], freq=x$Freq)
```

Testing and Training Data Sets

We need to make sure that we are not overfitting the data. Sometimes we want to separate the data set so that we can see how well our model fits the data on a new data set.

Once we have cleaned the Titanic data set, we can see that we have 2201 observations in our data frame. We can separate this into 75% training and 25% testing.

This is done as follows:

```
set.seed(101) # Set Seed so that same sample can be reproduced
# Now Selecting 75% of data as sample from total 'n' rows
sample <- sample.int(n = nrow(titanic),
                                         size =
round(.75*nrow(titanic),0),
                                         replace = F)
#getting the observations with the random values from the sample
train <- titanic[sample, ]
#eliminating the observations with random values from the sample
test <- titanic[-sample, ]
```

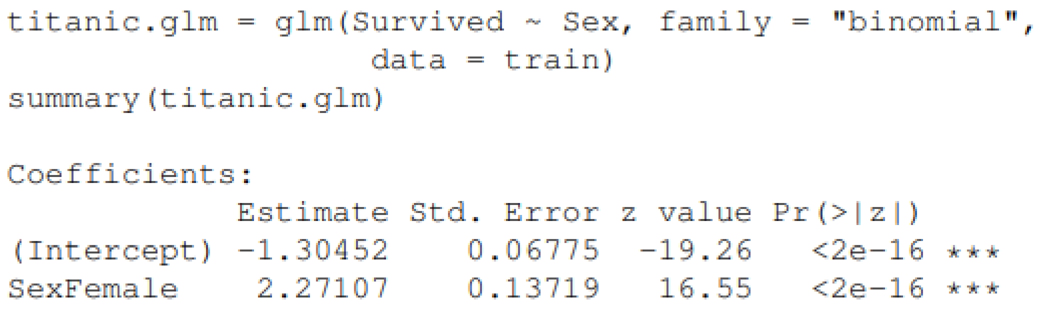
The sample array will be $0.75 * 2201 = 1650.75$, rounded to the whole number, 1651 values between 1 to 2201.

Model

(Recall Qualitative Predictors & Interaction Model)
The model will be as follows

$$P(\text{survived}|\text{Gender}) = p(X) = \begin{cases} \frac{\exp^{\beta_0}}{1+\exp^{\beta_0}} & \text{if Male} \\ \frac{\exp^{\beta_0+\beta_1}}{1+\exp^{\beta_0+\beta_1}} & \text{if Female} \end{cases}$$

In R:



```
titanic.glm = glm(Survived ~ Sex, family = "binomial",
                  data = train)
summary(titanic.glm)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.30452    0.06775  -19.26   <2e-16 ***
SexFemale    2.27107    0.13719   16.55   <2e-16 ***
```

thus:

$$\hat{p}(\text{survival}|\text{male}) = \frac{\exp(-1.30452)}{1+\exp(-1.30452)} = 0.202$$

and

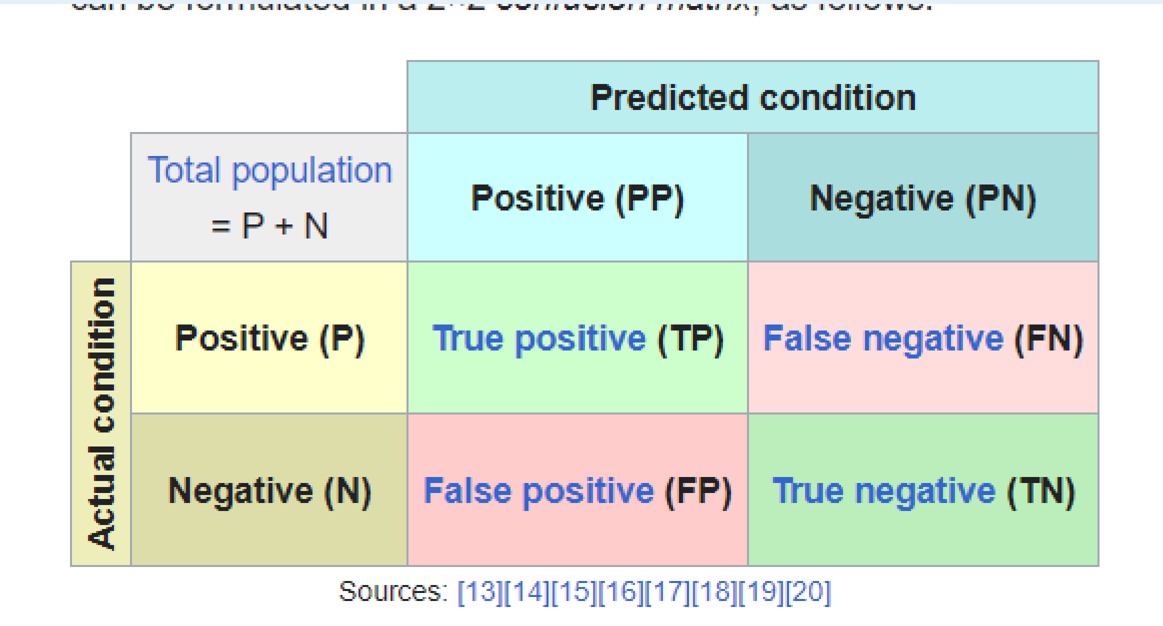
$$\hat{p}(\text{survival}|\text{female}) = \frac{\exp(-1.30452+2.27107)}{1+\exp(-1.30452+2.27107)} = 0.7308$$

of all the males, the males only have a 20% chance of surviving. Of all the population that is female, the females have a 73% chance of surviving.

Confusion Matrix

A confusion matrix is a convenient way to display to observations that are incorrectly assigned to the wrong category.

ⓘ **General view of the confusion matrix** ›



can be formulated in a 2×2 confusion matrix, as follows:

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

Sources: [13][14][15][16][17][18][19][20]

look where both the column label and row label match, that is the "true" output, otherwise its a "false" output.

The following table is the confusion matrix for the *training data.*

		True Survive	
		No	Yes
Predicted	No	1034	262
Survive	Yes	93	262

percent predicted correctly: $\frac{1034+262}{1651} = 0.785$
(generally: $\frac{\text{true\_positive}+\text{true\_negative}}{\text{sample\_size}}$)
(the sample size of the *training data* is 1651)

The following table is the confusion matrix for the *test data set.*

		True Survive	
		No	Yes
Predicted	No	330	105
Survive	Yes	33	82

The error rate is the percent that was wrongly or incorrectly predicted.
error rate of test data: $\frac{105+33}{550}$

(generally: $\frac{\text{false\_positive+false\_negative}}{\text{sample\_size}}$)
(the sample size of the *test data* is 550)

Sensitivity and Specificity

- **Sensitivity** measures the proportion of positives that are correctly identified. $\frac{\text{true\_positive}}{\text{false\_positive+true\_positive}}$
- **Specificity** measures the proportion of negatives that are correctly identified. $\frac{\text{true\_negative}}{\text{false\_negative+true\_negative}}$

for our test data:

$$\text{Sensitivity} = \frac{82}{105 + 82} = 0.4385$$

$$\text{Specificity} = \frac{330}{330 + 33} = 0.9091$$