Logistic Regression

Classification problems

- From now we are going to look at the classification problem.

- Classification problems are similar to regression models, with one important exception:
    - The dependent/predicted variable is categorical variable.

Logistic regression

- There are set of numeric and categorical independent variables
- The dependent variable is a binary variable with two possible categories/classes
  - These classes are usually called Positive/Negative, or Success/Failure, etc.
  - We can define which class is the Positive and which one is the Negative
  - The dependent variable follows Binomial distribution
- The goal is to predict the probability of the case to belong to one of the classes

Logistic regression formula:

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$$

where $logit(p)$ is the log of the odds

and p is the probability of success, or the probability of the case to be Positive

X are the independent variables

Odds

- If there is a 75% chance that it will rain tomorrow, then 3 out of 4 times we say this it will rain. That means for every three times it rains once it will not. The odds of it raining tomorrow are 3 to 1. This can also be understood as $(3/4)/1/4=3/1$.
- If the odds that the horse will win the race is 1 to 3, that means for every 4 races it runs, it will win 1 and lose 3.

Question!

Lets say during the last 20 games Betis won 9. What are the odds of winning for Betis?

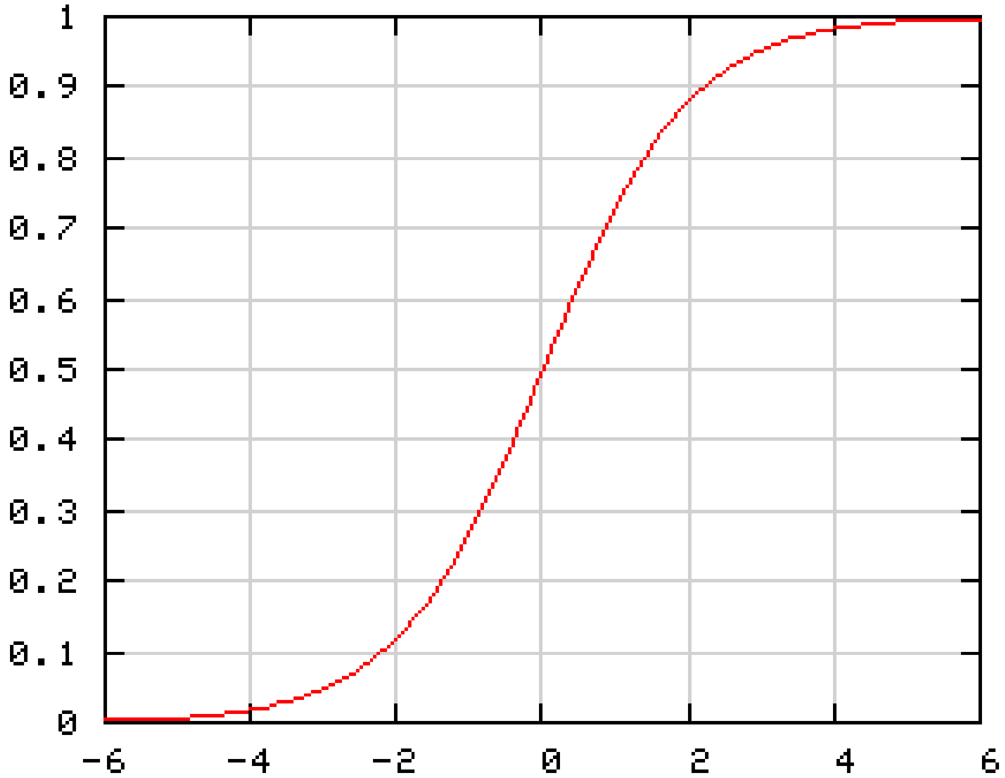
The probability of the case to be Positive is calculated with the following formula

$$P(Y_i) = \frac{e^{b_0+b_1X_1+b_2X_2+\ldots+b_kX_k}}{1 + e^{b_0+b_1X_1+b_2X_2+\ldots+b_kX_k}}$$

Where $b_0, b_1, b_2 \ldots b_k$ are coefficient estimates for $\beta_0, \beta_1, \beta_2 \ldots \beta_k$

Logistic regression

With the given function specificatins, the predicted probability is always going to be within the range [0:1]



Logistic Regression: Titanic case

Lets say we want to predict the probability of survival based on sex only

$$\ln\left(\frac{p_{surv}}{1-p_{surv}}\right) = \beta_0 + \beta_1 sex$$

Logistic Regression

First we will transform survived into binomial categorical variable
Please not, as alphabetically Yes comes after No, Yes will be treated as Positive case, No as negative case, No=0, Yes=1

```
Titanic<-read.csv("Titanic_imputed.csv")

Titanic$pclass<-as.factor(Titanic$pclass)
Titanic$survived<-factor(Titanic$survived, levels=c(0,1),
                         labels=c("No", "Yes"))
```

Logistic Regression

- glm stands for Generalized Linear Model
- Syntax is the same as with linear regression
- family="binomial" argument tells R that logistic regression needs to be fitted

```
model1<-glm(survived~sex, data=Titanic, family="binomial")
```

Logistic Regression

Note that female is the base/reference category, as alphabetically it comes first

```r
model1<-glm(survived~sex, data=Titanic, family="binomial")
summary(model1)
```

```
##
## Call:
## glm(formula = survived ~ sex, family = "binomial", data = Titanic)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6124  -0.6511  -0.6511   0.7977   1.8196
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.9818     0.1040   9.437   <2e-16 ***
## sexmale      -2.4254     0.1360 -17.832   <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: 1741.0  on 1308  degrees of freedom
## Residual deviance: 1368.1  on 1307  degrees of freedom
## AIC: 1372.1
##
## Number of Fisher Scoring iterations: 4
```

Logistic Regression

```r
coef(model1)
```

```
## (Intercept)     sexmale
##    0.981813   -2.425438
```

$$\ln\left(\frac{p_{surv}}{1-p_{surv}}\right) = \beta_0 + \beta_1 sex$$

β coefficient shows:

$$\ln\left(\frac{p_{surv\ males}}{1-p_{surv\ males}}\right) - \ln\left(\frac{p_{surv\ females}}{1-p_{surv\ females}}\right) = -2.42$$

Logistic Regression

```r
exp(coef(model1))
```

```
## (Intercept)     sexmale
##  2.66929134  0.08843935
```

$\exp(\beta)$ coefficient shows:

$$\exp\left(\ln\left(\frac{p_{surv\ males}}{1-p_{surv\ males}}\right)-\ln\left(\frac{p_{surv\ females}}{1-p_{surv\ females}}\right)\right)=\frac{\frac{p_{surv\ males}}{1-p_{surv\ males}}}{\frac{p_{surv\ females}}{1-p_{surv\ females}}}=0.088$$

$\frac{\frac{p_{surv\ males}}{1-p_{surv\ males}}}{\frac{p_{surv\ females}}{1-p_{surv\ females}}}$ is called Odds ratio

- The coefficient shows the change in the log odds for the one unit change in the independent variable.

$$\ln\left(\frac{P}{1-P}\right) males - \ln\left(\frac{P}{1-P}\right) female$$

- To understand the change in the odds ratio, we need to take the exponential of the coefficient

$$\exp(\beta) = \frac{\frac{P}{1-P} males}{\frac{P}{1-P} female}$$

Logistic Regression

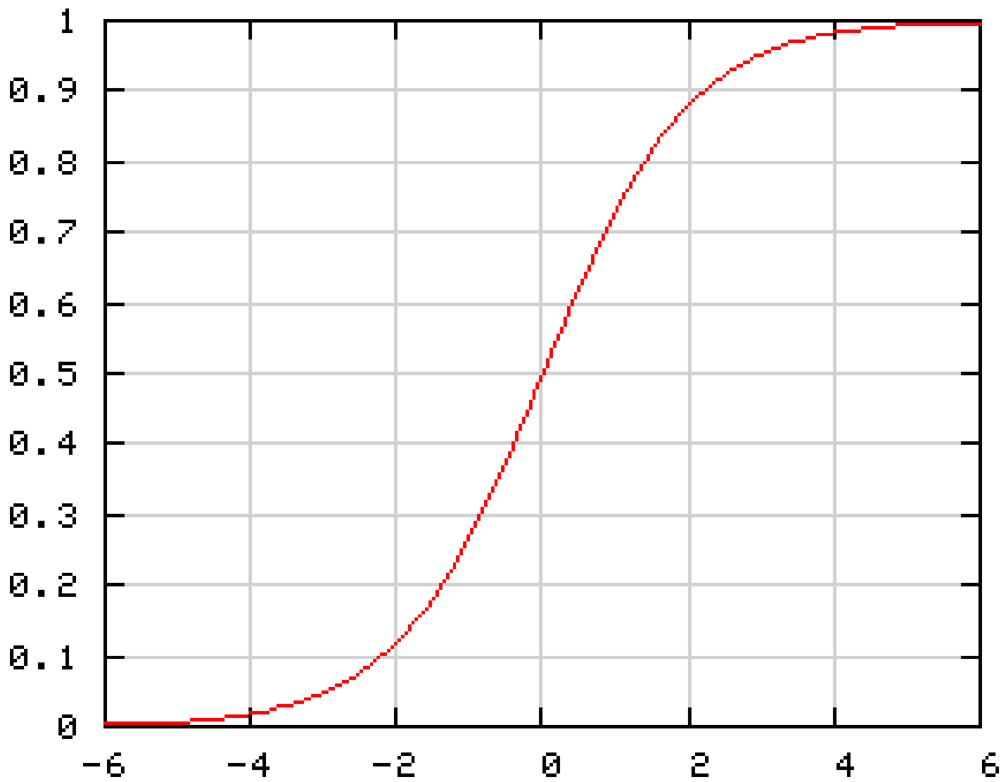
So if you change the gender from female to male, than the odds ratio of survival will decrease by 11 times (1/0.088).

OR: Odds of survival for male is 8.8% of the odds of survival for female

Conclusion: Females likelihood to survive is 11 times more than for males

Logistic Regression

Coefficients explain effect of the independent variable on the logits and odds ratio and not the probability by itself.



Logistic Regression

By hand

```r
table(Titanic$sex, Titanic$survived)
```

```
##
##           No Yes
##   female 127 339
##   male   682 161
```

```r
addmargins(table(Titanic$sex, Titanic$survived))
```

```
##
##            No  Yes  Sum
##   female  127  339  466
##   male    682  161  843
##   Sum     809  500 1309
```

Logistic Regression

```r
addmargins(table(Titanic$sex, Titanic$survived))
```

```
##
##              No  Yes  Sum
##    female   127  339  466
##    male     682  161  843
##    Sum      809  500 1309
```

$P(S|M) - Probability\ of\ survival\ for\ males$

$Odds(Male)$

$P(S|F)\ Probability\ of\ survival\ for\ females$

$Odds(female)$

Logistic Regression

```r
addmargins(table(Titanic$sex, Titanic$survived))
```

```
##
##             No  Yes  Sum
##   female   127  339  466
##   male     682  161  843
##   Sum      809  500 1309
```

$$P(S|M) - \frac{161}{843} = 0.19$$

$$P(S|F) = \frac{339}{466} = 0.72$$

$$Odds(Male) = \frac{0.19}{1-0.19} = 0.24$$

$$Odds(female) = \frac{0.72}{1 - 0.72} = 2.57$$

$$odds\ ratio = \frac{0.24}{2.57} = 0.09 \approx 0.088\text{, our regression coefficient}$$

Logistic Regression

Adding more variables: pclass (Passenger class), Age and sibsp (number of siblings traveling with)

- Passenger class is categorical variable, 1st class (riches people) ----3rd class (poorest people)

- sibsp is numeric variable

Logistic Regression

```r
model2<-glm(survived~sex+pclass+age+sibsp,data=Titanic, family ="binomial")
# Summary of the model
summary(model2)
```

```
##
## Call:
## glm(formula = survived ~ sex + pclass + age + sibsp, family = "binomial",
##     data = Titanic)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4448  -0.6717  -0.4331   0.6736   2.4817
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.291020   0.279306  11.783  < 2e-16 ***
## sexmale     -2.563299   0.152056 -16.858  < 2e-16 ***
## pclass2     -1.112127   0.206785  -5.378 7.52e-08 ***
## pclass3     -2.079510   0.191969 -10.833  < 2e-16 ***
## age         -0.026882   0.005241  -5.129 2.91e-07 ***
## sibsp       -0.300326   0.081834  -3.670 0.000243 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1741.0  on 1308  degrees of freedom
## Residual deviance: 1219.9  on 1303  degrees of freedom
## AIC: 1231.9
##
## Number of Fisher Scoring iterations: 4
```

How will you interpret the exponents of the coefficients ?

```
exp(coef(model2))
```

```
## (Intercept)     sexmale     pclass2     pclass3         age       sibsp
## 26.87024936  0.07705016  0.32885860  0.12499140  0.97347567  0.74057647
```

Logistic Regression

```
exp(coef(model2))
```

```
## (Intercept)     sexmale     pclass2     pclass3         age       sibsp
## 26.87024936  0.07705016  0.32885860  0.12499140  0.97347567  0.74057647
```

- sex: The odds of survival for males is 7.7% of females
- pclass2 and pclass3 both are categories of pclass variable, with pclass1 being the base/reference category
    - Passengers of second class had 68% less odds to survive compared to passengers of class 1
    - Passengers of third class had 88% less odds to survive compared to passengers of class 1
- Age: one unit increase in age decreases the odds ratio of survival by 3% (1-0.973)
- sibsp: 1 unit increase in number of siblings a person is traveling with, decreases the odds to survive by 26%

Predict probability for the single case

Person: Gender=Female, Age=20, pclass=2, sibsp=2

$$P(Y_i) = \frac{e^{b_0+b_1X_1+b_2X_2+\ldots+b_kX_k}}{1+e^{b_0+b_1X_1+b_2X_2+\ldots+b_kX_k}}$$

use exp() for exponent

Predict probability for the single case

Person: Gender=Female, Age=17, pclass=1, sibsp=2

$$P(Y_i) = \frac{e^{b_0+b_1X_1+b_2X_2+\ldots+b_kX_k}}{1+e^{b_0+b_1X_1+b_2X_2+\ldots+b_kX_k}}$$

```r
coef(model2)
```

```
## (Intercept)     sexmale     pclass2     pclass3         age       sibsp
##  3.29101970 -2.56329858 -1.11212741 -2.07951031 -0.02688245 -0.30032638
```

```r
exp(3.29101970+ -2.56329858*0+-1.11212741*0 + -0.02688245*17
               +-0.30032638*2)/(1+exp(3.29101970+ -2.56329858*0+-1.11212741*0
                                       + -0.02688245*17+-0.30032638*2))
```

```
## [1] 0.903206
```

Case 2. {male, pclass3, age=20, sibsp=0}

Football data

Result: Have a value of 1 if home team won and value of 0 otherwise
FTHG: Number of the goals scored by home team

```
seriea <- read.csv("seriea_games.csv")
head(seriea)
```

```
##        DATE HOMETEAM   AWAYTEAM FTHG FTAG FTTG Result
## 1 8/20/2016 Juventus Fiorentina    2    1    3      1
## 2 8/20/2016     Roma    Udinese    4    0    4      1
## 3 8/21/2016 Atalanta      Lazio    3    4    7      0
## 4 8/21/2016  Bologna    Crotone    1    0    1      1
## 5 8/21/2016   Chievo      Inter    2    0    2      1
## 6 8/21/2016   Empoli  Sampdoria    0    1    1      0
```

Interpret the coefficient for FTHG

```r
fmod <- glm(Result~FTHG, data=seriea, family='binomial')
summary(fmod)
```

```
##
## Call:
## glm(formula = Result ~ FTHG, family = "binomial", data = seriea)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4250  -0.7419  -0.2873   0.4565   1.6880
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.1671     0.2299  -13.78   <2e-16 ***
## FTHG          2.0177     0.1429   14.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

Frequency table

```
t <-table(seriea$FTHG, seriea$Result)
t
```

```
##
##          0    1
##     0  178    0
##     1  173   70
##     2   52  120
##     3    9   95
##     4    0   41
##     5    0   14
##     6    0    5
##     7    0    3
```

Create a dataframe

```r
Wins <- t[,2]
df <- data.frame(Goals = 0:7, Wins)
df
```

```
##   Goals Wins
## 0     0    0
## 1     1   70
## 2     2  120
## 3     3   95
## 4     4   41
## 5     5   14
## 6     6    5
## 7     7    3
```

```r
prop.table(t,1)
```

```
##
##            0          1
##   0 1.00000000 0.00000000
##   1 0.71193416 0.28806584
##   2 0.30232558 0.69767442
##   3 0.08653846 0.91346154
##   4 0.00000000 1.00000000
##   5 0.00000000 1.00000000
##   6 0.00000000 1.00000000
##   7 0.00000000 1.00000000
```

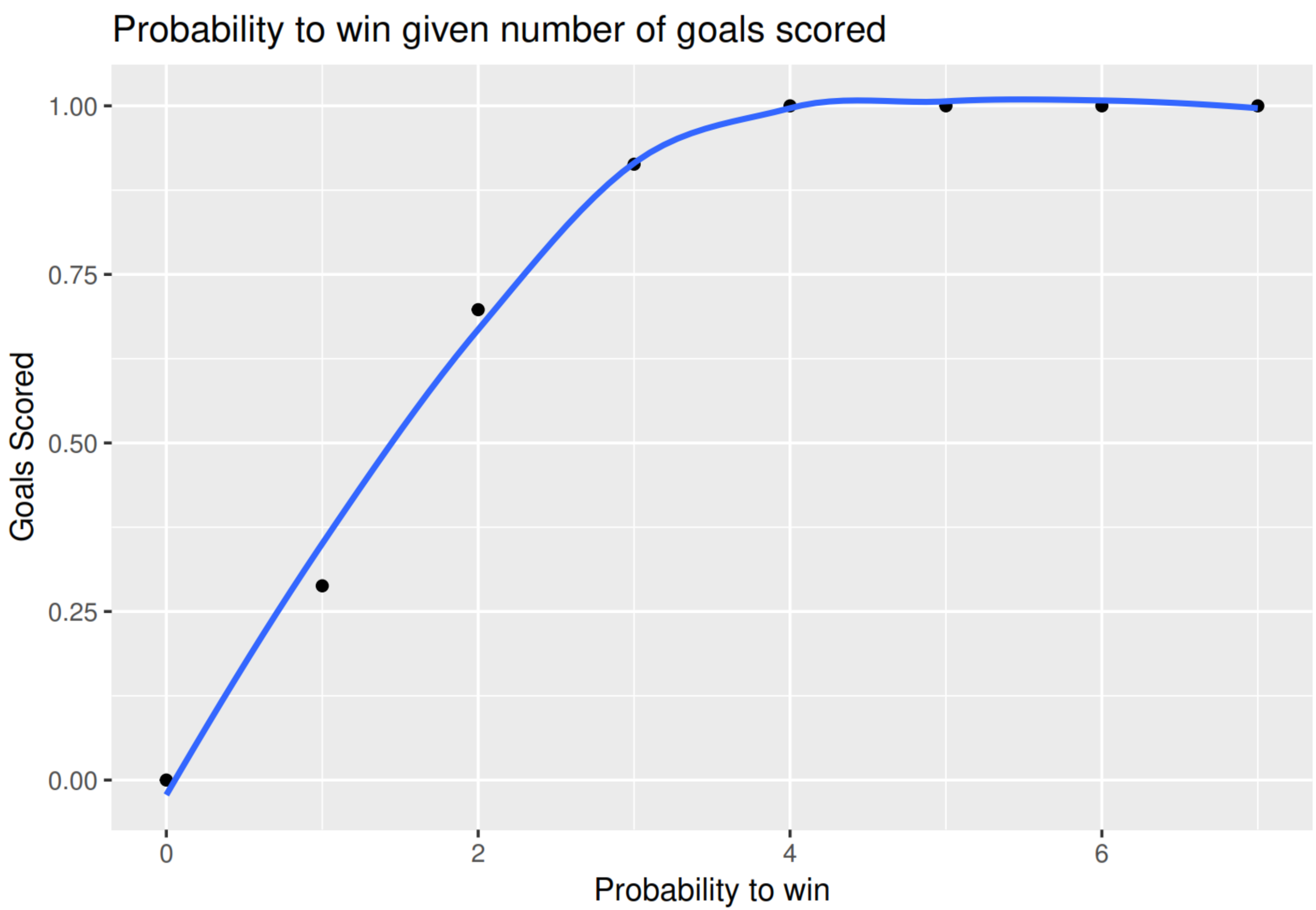
If the home score one goal, probability of wining is 0.6976

```r
df$Probs <- prop.table(t,1)[,2]
df
```

```
##   Goals Wins     Probs
## 0     0    0 0.0000000
## 1     1   70 0.2880658
## 2     2  120 0.6976744
## 3     3   95 0.9134615
## 4     4   41 1.0000000
## 5     5   14 1.0000000
## 6     6    5 1.0000000
## 7     7    3 1.0000000
```

Sigmoid curve

```r
ggplot(df, aes(x=Goals, y=Probs))+geom_point()+geom_smooth(se=F)+
  labs(x="Probability to win", y="Goals Scored",
       title="Probability to win given number of goals scored")
```



Testing model performance

Naïve Rule

Naïve rule: classify all records as belonging to the most prevalent class

- Often used as benchmark: we hope to do better than that
- Exception: when goal is to identify high-value but rare outcomes, we may do well by doing worse than the naïve rule.

Cutoff for classification

Most DM algorithms classify via a 2-step process:

For each record,

1. Compute **probability of belonging to class "1"**
2. Compare to cutoff value, and classify accordingly

- Default cutoff value is 0.50

  If >= 0.50, classify as "1"

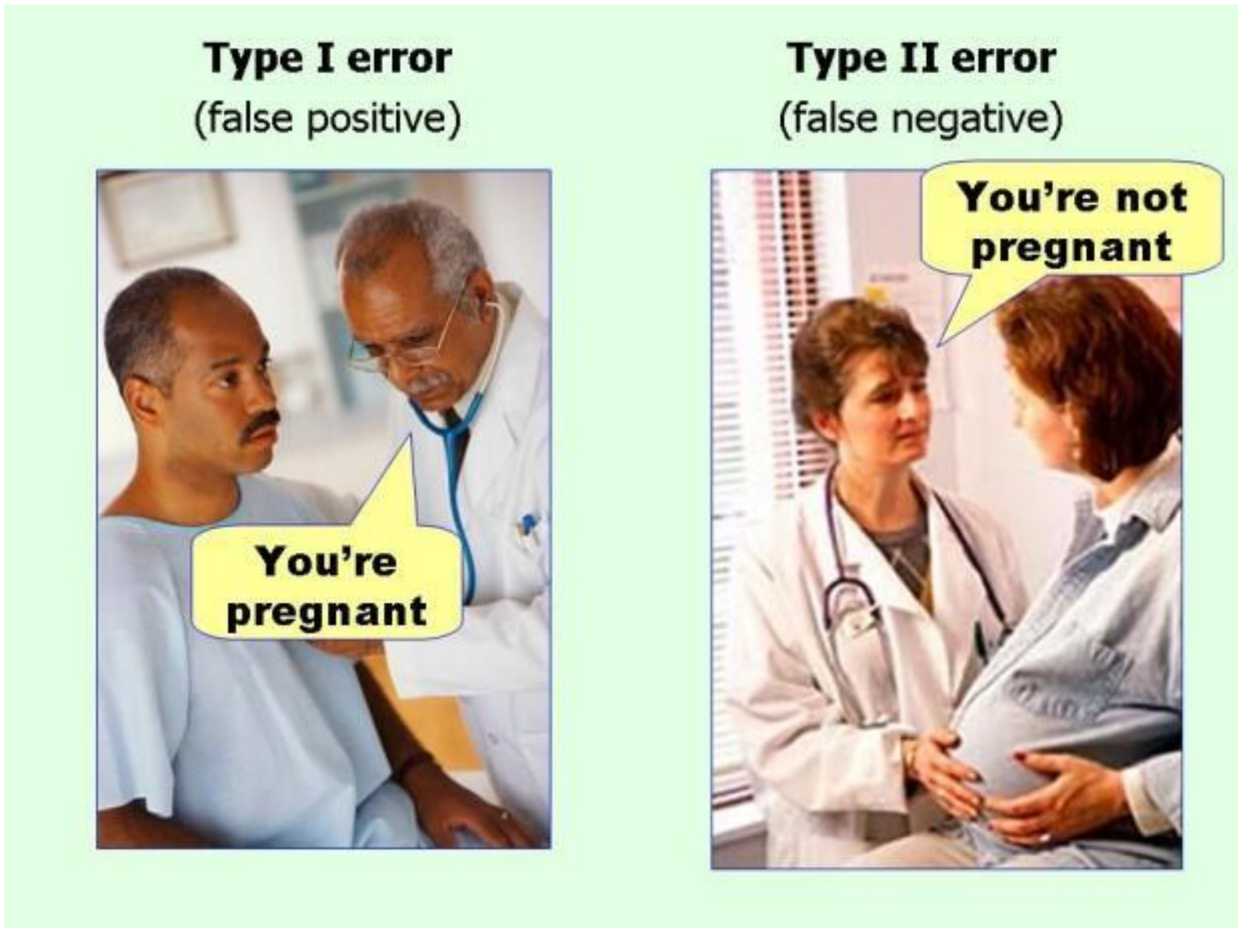
  If < 0.50, classify as "0"

- Can use different cutoff values
- Typically, error rate is lowest for cutoff = 0.50

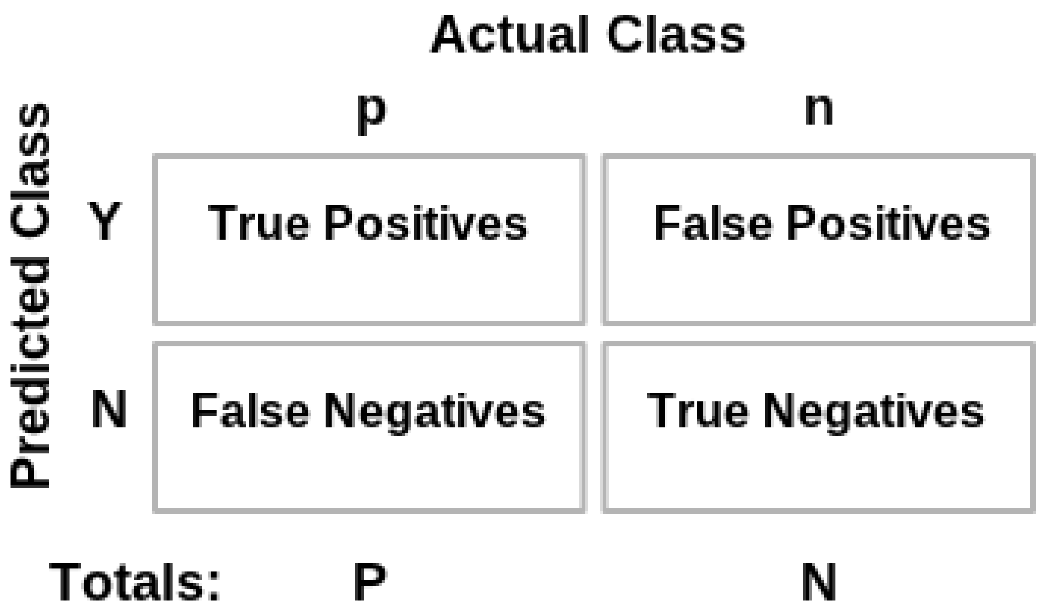
Confusion matrix

		Actual Class	
		p	n
Predicted Class	Y	True Positives	False Positives
	N	False Negatives	True Negatives
Totals:		P	N

False Positive and False Negative

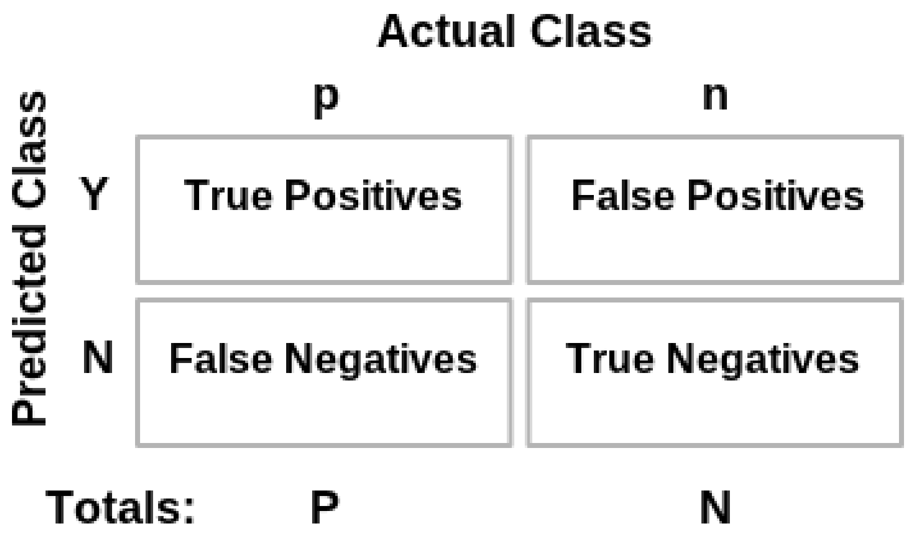


Confusion matrix



Overall accuracy= (True Positives + True Negatives/(True Positives + False Negatives+ False Positives+ True negatives)

Confusion matrix-Other measures of accuracy



Sensitivity= True Positives/(True Positives + False Negatives)

Specificity= True Negatives/(True Negatives + False Positives)

Positive Predictive Value= True Positives/(True Positives + False Positives)

Negative Predictive Value= True Negatives/(True Negatives + False Negatives)

Example

The logistic regression is used to predict the court decision (Guilty=1/ Not Guilty=0)

Predicted class		Actual class: Guilty	Actual class: Not guilty
	Guilty	20	5
	Not Guilty	9	25

Example

The logistic regression is used to predict the court decision (Guilty=1/ Not Guilty=0)

Predicted class \ Actual class	Guilty	Not guilty
Guilty	20	5
Not Guilty	9	25

Sensitivity- Given that someone is actually guilty, what is the probability that the model will make correct decision

$$P(Predicts\ guilty\ |Actually\ Guilty) = \frac{20}{29} = 68\%$$

Example

The logistic regression is used to predict the court decision (Guilty=1/ Not Guilty=0)

		Actual class	
		Guilty	Not guilty
Predicted class	Guilty	20	5
	Not Guilty	9	25

Example

The logistic regression is used to predict the court decision (Guilty=1/ Not Guilty=0)

		Actual class	
		Guilty	Not guilty
Predicted class	Guilty	20	5
	Not Guilty	9	25

Specificity- Given that someone is actually not guilty, what is the probability that the model will classify him as not guilty

$$P(Predicts\ not\ guilty\ |Actualy\ Not\ Guilty) = \frac{25}{30} = 83\%$$

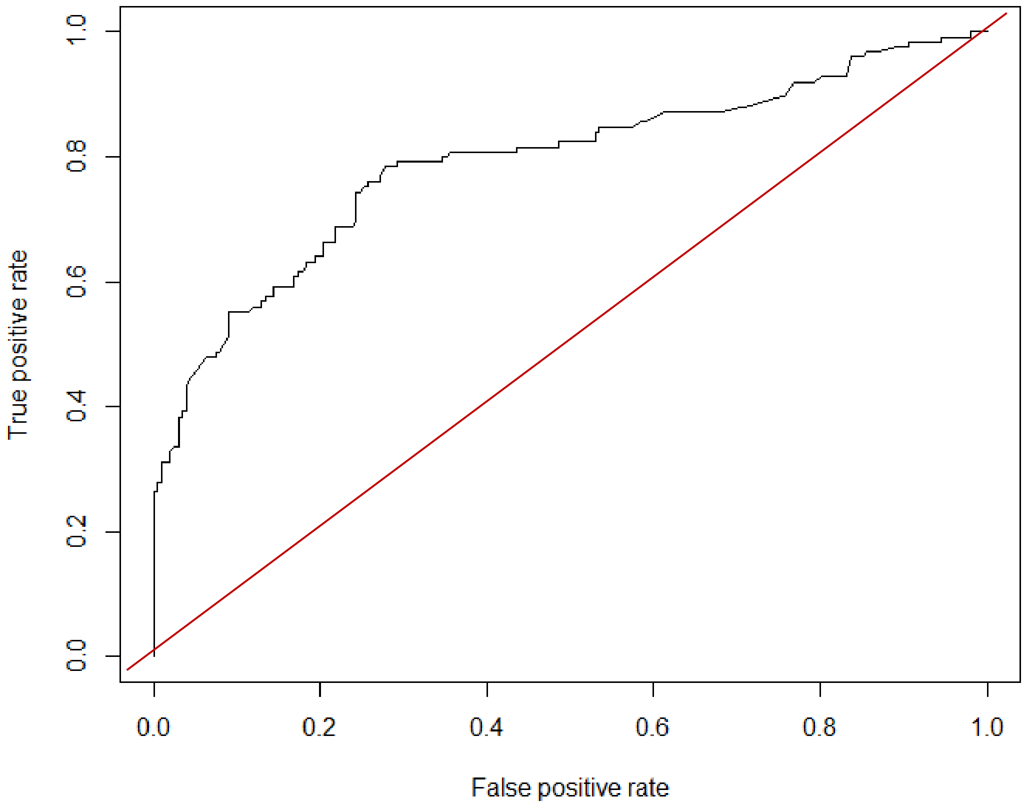
Cutoff Table

Actual Class	Prob. of "1"	Actual Class	Prob. of "1"
1	0.996	1	0.506
1	0.988	0	0.471
1	0.984	0	0.337
1	0.980	1	0.218
1	0.948	0	0.199
1	0.889	0	0.149
1	0.848	0	0.048
0	0.762	0	0.038
1	0.707	0	0.025
1	0.681	0	0.022
1	0.656	0	0.016
0	0.622	0	0.004

- If cutoff is 0.50: eleven records are classified as “1”
- If cutoff is 0.80: seven records are classified as “1”

ROC Curve

- Models accuracy can change if you change the cut off value.
- The trade-off between True Positive rate (Sensitivity) and False Positive Rate (1-Specificity) for the different cut-off values is given by ROC curve.



ROC curve

1.It shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).

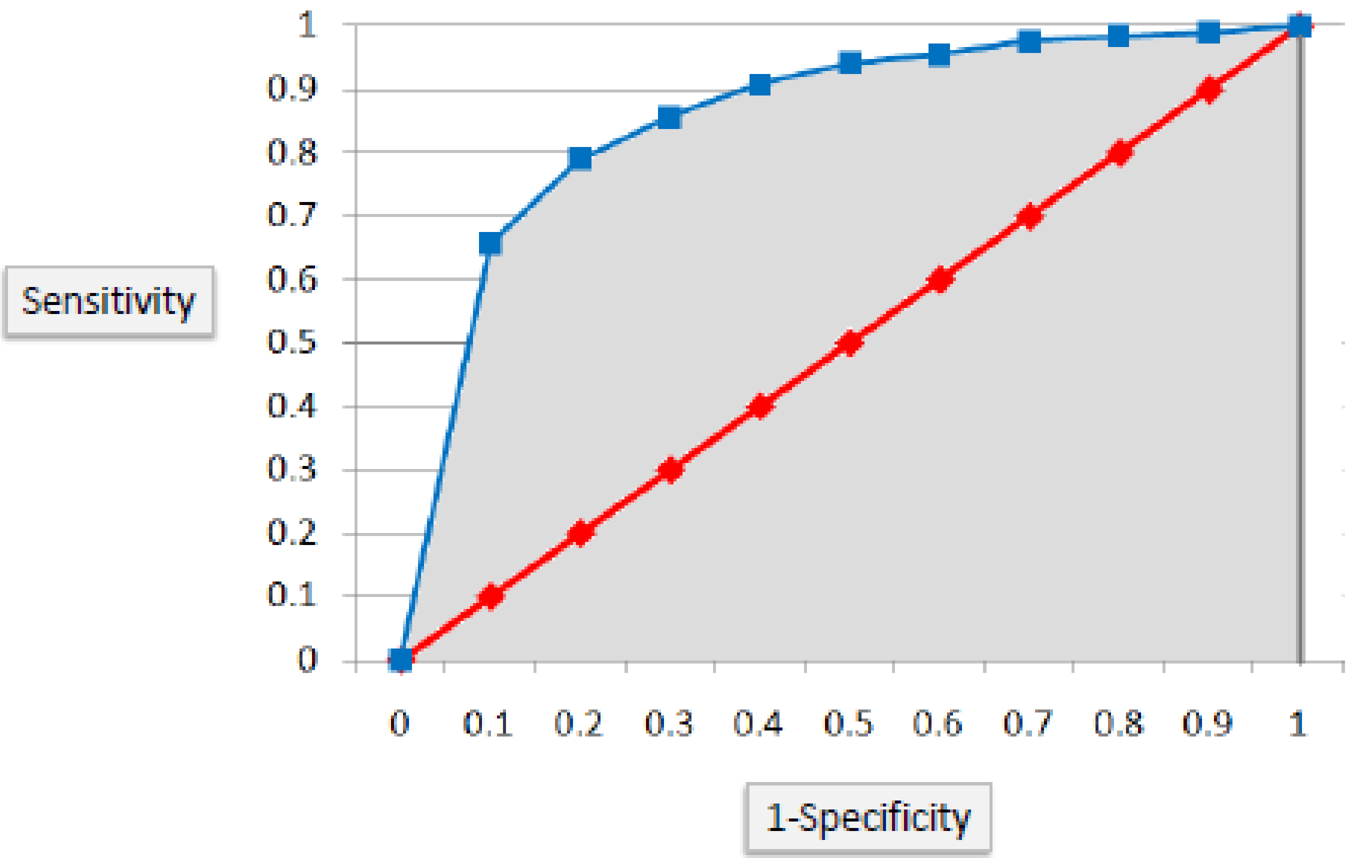
2.The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the model.

3.The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate is the model.

4. The random guess model will have ROC curve on the diagonal

Area under the curve

Is in the range of [0:1]. Higher value indicates better model performance (Higher Accuracy). **The random guess model has AUC of 0.5 area under the red line).**



Development of ROC curve

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	p	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	10	n	.1

Fill the confusion matrix

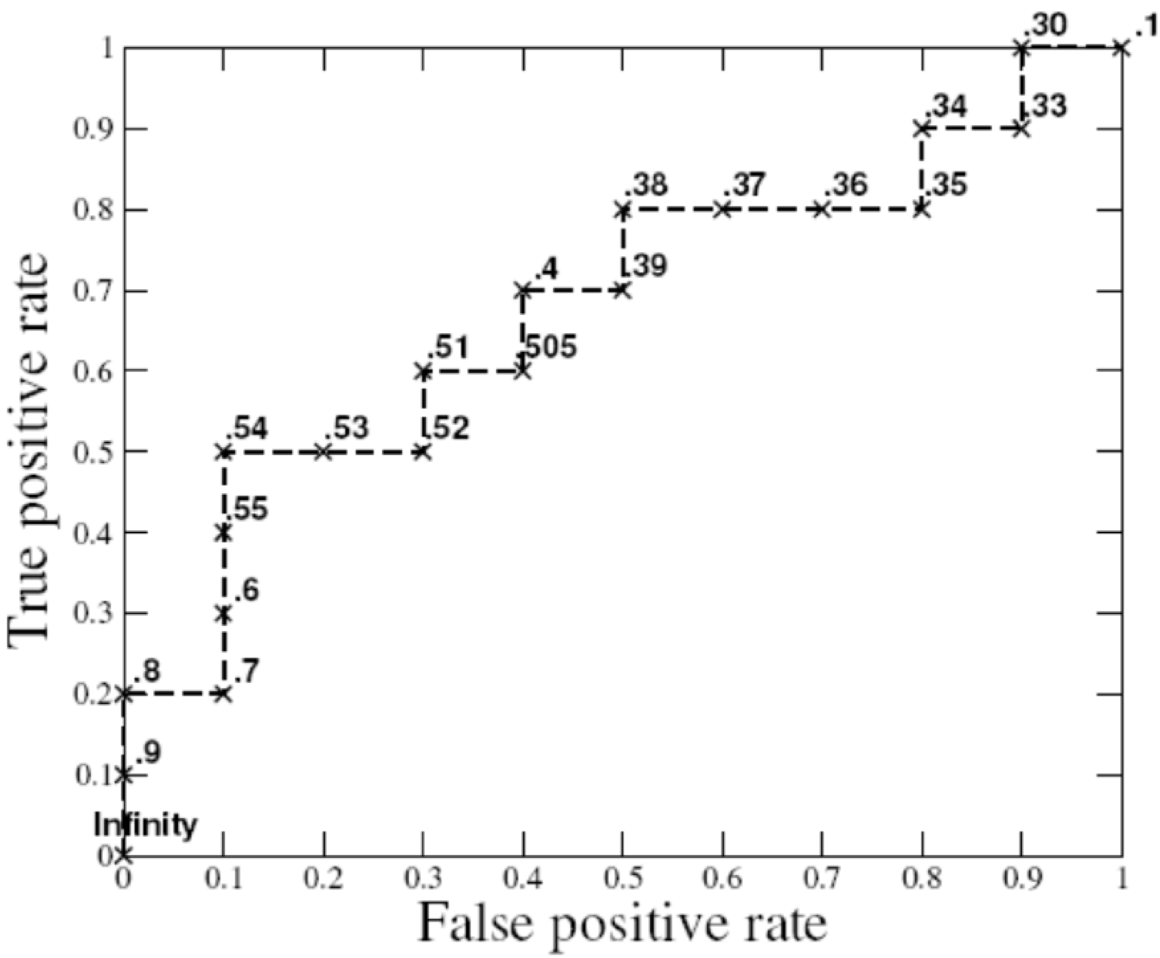
Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	p	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	10	n	.1

Cut-off=0.54

True Positive Rate (Sensitivity) - ?
False Positive Rate (1-Specificity) - ?

	Actual		
Predicted	Positive	Negative	***Total***
Positive			
Negative			
Total			

The ROC curve



Doing in R

Doing in R

```r
credit<-read.csv("Credit.csv")
str(credit)
```

```
## 'data.frame':    700 obs. of  9 variables:
##  $ age     : int  41 27 40 41 24 41 39 43 24 36 ...
##  $ ed      : Factor w/ 5 levels "college degree",..: 1 3 3 3 2 2 3 3 3 3 ...
##  $ employ  : int  17 10 15 15 2 5 20 12 3 0 ...
##  $ address : int  12 6 14 14 0 5 9 11 4 13 ...
##  $ income  : int  176 31 55 120 28 25 67 38 19 25 ...
##  $ debtinc : num  9.3 17.3 5.5 2.9 17.3 10.2 30.6 3.6 24.4 19.7 ...
##  $ creddebt: num  11.359 1.362 0.856 2.659 1.787 ...
##  $ othdebt : num  5.009 4.001 2.169 0.821 3.057 ...
##  $ default : Factor w/ 2 levels "No","Yes": 2 1 1 1 2 1 1 1 2 1 ...
```

Doing in R

Create data partition with package caret

Training and testing sets need to have the same proportions of the classes in dependent variable. createDataPartition function does it.

```
prop.table(table(credit$default))
```

```
##
##        No       Yes
## 0.7385714 0.2614286
```

Divide to training and testing sets, 80/20

```
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
set.seed(1)
trainIndex <- createDataPartition(credit$default,
                                  p = .8, list = FALSE)
Train<-credit[trainIndex,]
Test<-credit[-trainIndex,]
```

Doing in R

The model

```
credit1<-glm(default~., data=Train, family="binomial")
summary(credit1)
```

```
##
## Call:
## glm(formula = default ~ ., family = "binomial", data = Train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2635  -0.6624  -0.2974   0.3527   2.9164
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.327700   0.721456  -1.840   0.0657 .
## age               0.042901   0.019471   2.203   0.0276 *
## edhigh school    -0.341689   0.396523  -0.862   0.3888
## edno high school -0.363650   0.377390  -0.964   0.3353
## edpostgraduate    0.370586   1.463332   0.253   0.8001
## edundergraduate  -0.858867   0.569879  -1.507   0.1318
## employ           -0.261398   0.035950  -7.271 3.57e-13 ***
## address          -0.105268   0.025351  -4.152 3.29e-05 ***
## income           -0.005308   0.008603  -0.617   0.5372
## debtinc           0.061331   0.033183   1.848   0.0646 .
## creddebt          0.580753   0.120869   4.805 1.55e-06 ***
## othdebt           0.083504   0.084553   0.988   0.3233
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 645.34  on 560  degrees of freedom
## Residual deviance: 450.40  on 549  degrees of freedom
## AIC: 474.4
##
## Number of Fisher Scoring iterations: 6
```

Doing in R

use argument type="response" to get vector of probabilities for positive class

```
# Use type response to get predicted probabilities for Positive class "Yes"
pr1<-predict(credit1, newdata=Test, type="response")
pr1[1:50]
```

```
##           2           4           6           7          11          12
## 0.147173252 0.015163268 0.300966081 0.258438943 0.376250149 0.224048227
##          13          17          18          30          42          44
## 0.011591889 0.205221418 0.001192803 0.592613943 0.028699214 0.029959528
##          56          57          63          66          73          78
## 0.788467760 0.101403720 0.539791031 0.927217015 0.006179630 0.238288942
##          82          86          90          93          99         100
## 0.155037650 0.175591881 0.902924577 0.423779649 0.155236579 0.483559429
##         112         118         119         129         132         139
## 0.296477952 0.006581446 0.670644461 0.195736171 0.221626909 0.050220340
##         146         160         163         165         170         175
## 0.230358150 0.054655970 0.401267224 0.478071955 0.012009346 0.072038776
##         178         180         182         183         191         197
## 0.132136325 0.666839259 0.035547098 0.004600852 0.402006692 0.389522354
##         200         203         208         211         226         245
## 0.728316563 0.107395785 0.059784312 0.089108086 0.172862617 0.608317675
##         262         270
## 0.169055949 0.002267954
```

Doing in R

Lets make confusion matrix

```
table(Test$default, pr1>0.5)
```

```
##
##        FALSE TRUE
##    No     93   10
##    Yes    15   21
```

```
addmargins(table(Test$default, pr1>0.5))
```

```
##
##        FALSE TRUE Sum
##    No     93   10 103
##    Yes    15   21  36
##    Sum   108   31 139
```

Doing in R

Or predict the class label (NO, YES), with cut-off value of 0.5

```
pr_class<-factor(ifelse(pr1>0.5, "Yes", "No"))
```

```
addmargins(table(Test$default, pr_class))
```

```
##        pr_class
##          No Yes Sum
##    No    93  10 103
##    Yes   15  21  36
##    Sum  108  31 139
```

Calculate Overall Accuracy, Sensitivity, Specificity, NPV, PPV, (slide 31)

Doing in R

Calculate the accuracy

```
# Overall accuracy
(93+21)/(93+21+15+10)
```

```
## [1] 0.8201439
```

```
# Senstivity TP/(TP+FN) what percentage of Positive class is actually predicted correctly
21/(21+15)
```

```
## [1] 0.5833333
```

```
# Specificity TN/(TN+TP) what percentage of Negative classes is predicted correctly
93/(93+10)
```

```
## [1] 0.9029126
```

```
# Positive predictive value: TP/(TP+FP), the probability that if we
# predict the case to be positive it is going to be positive
21/(21+10)
```

```
## [1] 0.6774194
```

```
# Negative predictive value: TN(TN+FN), the probability that if we
# predict the case to be negative it is actually going to be negative
93/(93+15)
```

```
## [1] 0.8611111
```

Doing in R

You can use function confusionMatrix function from package caret.
The first argument is the predicted class, second is the actual class, and you need to specify which one is the positive case

```r
caret::confusionMatrix(pr_class, Test$default, positive="Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No Yes
##        No  93  15
##        Yes 10  21
##
##                Accuracy : 0.8201
##                  95% CI : (0.7461, 0.8801)
##     No Information Rate : 0.741
##     P-Value [Acc > NIR] : 0.01823
##
##                   Kappa : 0.5093
##  Mcnemar's Test P-Value : 0.42371
##
##             Sensitivity : 0.5833
##             Specificity : 0.9029
##          Pos Pred Value : 0.6774
##          Neg Pred Value : 0.8611
##              Prevalence : 0.2590
##          Detection Rate : 0.1511
##    Detection Prevalence : 0.2230
##       Balanced Accuracy : 0.7431
##
##        'Positive' Class : Yes
##
```

Doing in R

```
##
##                Accuracy : 0.8201
##                  95% CI : (0.7461, 0.8801)
##     No Information Rate : 0.741
##     P-Value [Acc > NIR] : 0.01823
##
```

- No Information rate: is your baseline prediction or the probability of the prevalent class:

- P-Value [Acc > NIR] - Test the hypothesis that the Accuracy is greater than No Information rate (lower p-value is better)

Doing in R

Package ROCR has a lot of handy tools for model performance evaluation.
1-step: Make a prediction object. The first argument is the predicted probabilities (not class labels), second argument is a vector with actual class labels.
2-step: Make a performance object, the argument is another object from step 1. Here you specify what you want to be on X and Y axes.

```
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

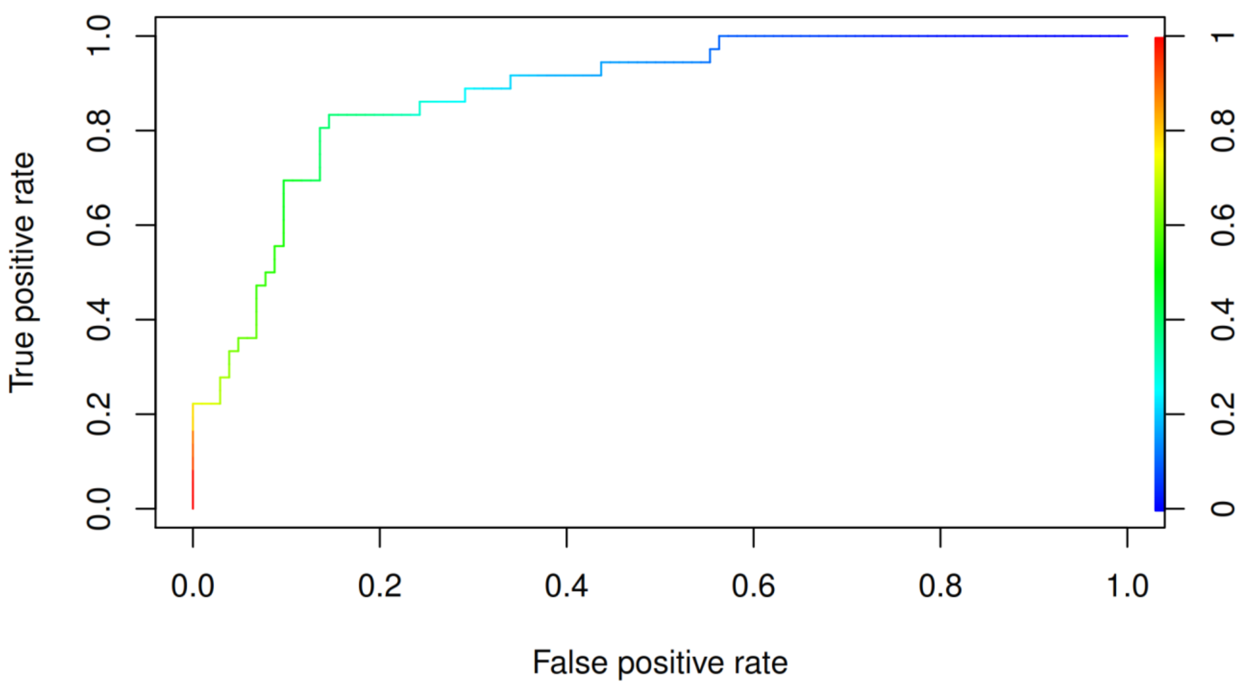
```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
P_Test <- prediction(pr1, Test$default)

perf <- performance(P_Test,"tpr","fpr")
```

Doing in R

```r
plot(perf, colorize=T)
```



Plot ROC curve and print AUC

```r
performance(P_Test, "auc")@y.values
```

```
## [[1]]
## [1] 0.8802589
```

- Lets build ROC curve using ggplot
- First we need to get FPR, TPR and cutoff values

```
str(perf)
```

```
## Formal class 'performance' [package "ROCR"] with 6 slots
##   ..@ x.name      : chr "False positive rate"
##   ..@ y.name      : chr "True positive rate"
##   ..@ alpha.name  : chr "Cutoff"
##   ..@ x.values    :List of 1
##   .. ..$ : num [1:140] 0 0 0 0 0 ...
##   ..@ y.values    :List of 1
##   .. ..$ : num [1:140] 0 0.0278 0.0556 0.0833 0.1111 ...
##   ..@ alpha.values:List of 1
##   .. ..$ : num [1:140] Inf 0.996 0.991 0.927 0.903 ...
```

Create a new dataframe with the data we need

```
FPR <- unlist(perf@x.values)
TPR <- unlist(perf@y.values)
alpha = unlist(perf@alpha.values)

df <- data.frame(FPR, TPR, alpha)
head(df)
```

```
##   FPR        TPR     alpha
## 1   0 0.00000000       Inf
## 2   0 0.02777778 0.9962384
## 3   0 0.05555556 0.9906777
## 4   0 0.08333333 0.9272170
## 5   0 0.11111111 0.9029246
## 6   0 0.13888889 0.8783368
```

Lets calculate by hand for the forth case

```
##    FPR        TPR     alpha
## 1    0 0.00000000       Inf
## 2    0 0.02777778 0.9962384
## 3    0 0.05555556 0.9906777
## 4    0 0.08333333 0.9272170
```

```
table(Test$default, pr1>0.9272170)
```

```
##
##        FALSE TRUE
##   No     103    0
##   Yes     33    3
```

```
# TPR
3/(3+33)
```

```
## [1] 0.08333333
```

```
# FPR
1-103/(103+0)
```

```
## [1] 0
```

```r
ggplot(df, aes(x=FPR, y=TPR, color=alpha))+geom_line()+
  theme_bw()+labs(x= "False Positive Rate", y= 'True Positive Rate',
                  title='ROC Curve', color="Cutoff values")
```