Logistic Regression

Jerome Reboulleau[1]

[1] HEC, Lausanne

Quantitative Method for Management

Outline

PART I:
LOGISTIC REGRESSION OVERVIEW

Logistic Regression

A specific regression

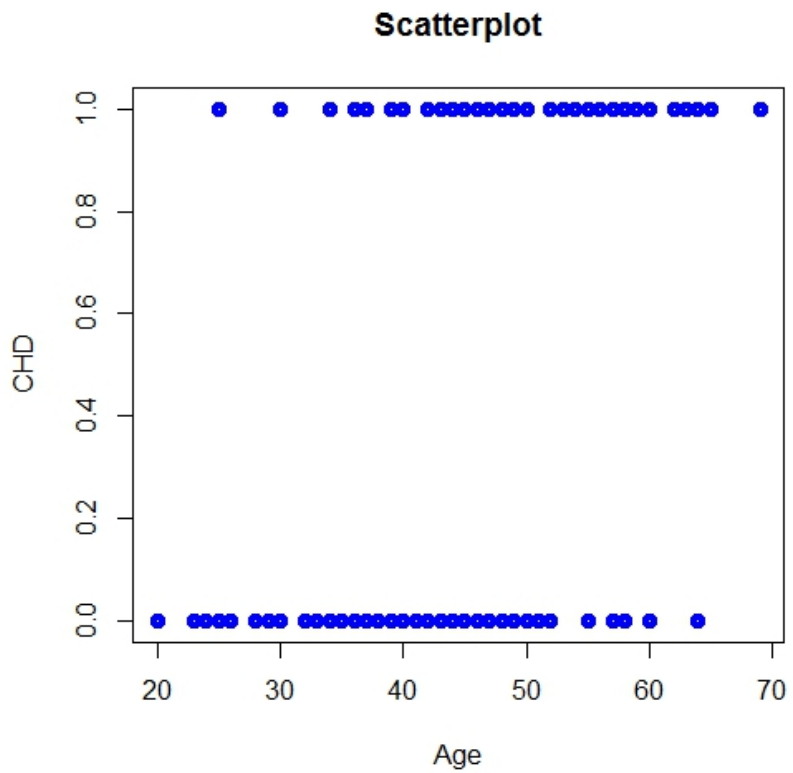
- The logistic regression is a specific case where the **outcome variable** is a **binary or dichotomous**.
- As a result, the methods employed will be very close to the one presented in the linear regression.
- We will use our Regression chapter as a motivation in the logistic regression context.
- We start by highlighting the differences between these 2 types of regression.

Example: Developing a Logistic Regression Model

- We consider the sample of heart diseases.
- We select 100 subjects (*ID*)and their age (*Age*).
- We note the presence or absence of significant coronary heart disease (*CHD*): 0 if absent, 1 if present.
- We also add an age group variable *AGRP*.
- The interest of the study is to explore the relationship between the Age and CHD.
- CHD is going to be our response variable.

Example: Developing a Logistic Regression Model

- Step 1: Scatterplot We start by developing an intuitive analysis of the various relationship:



- All points fall into 2 lines: $y = 0$ and $y = 1$.
- The structure of the relationship between CHD and Age is not clear.

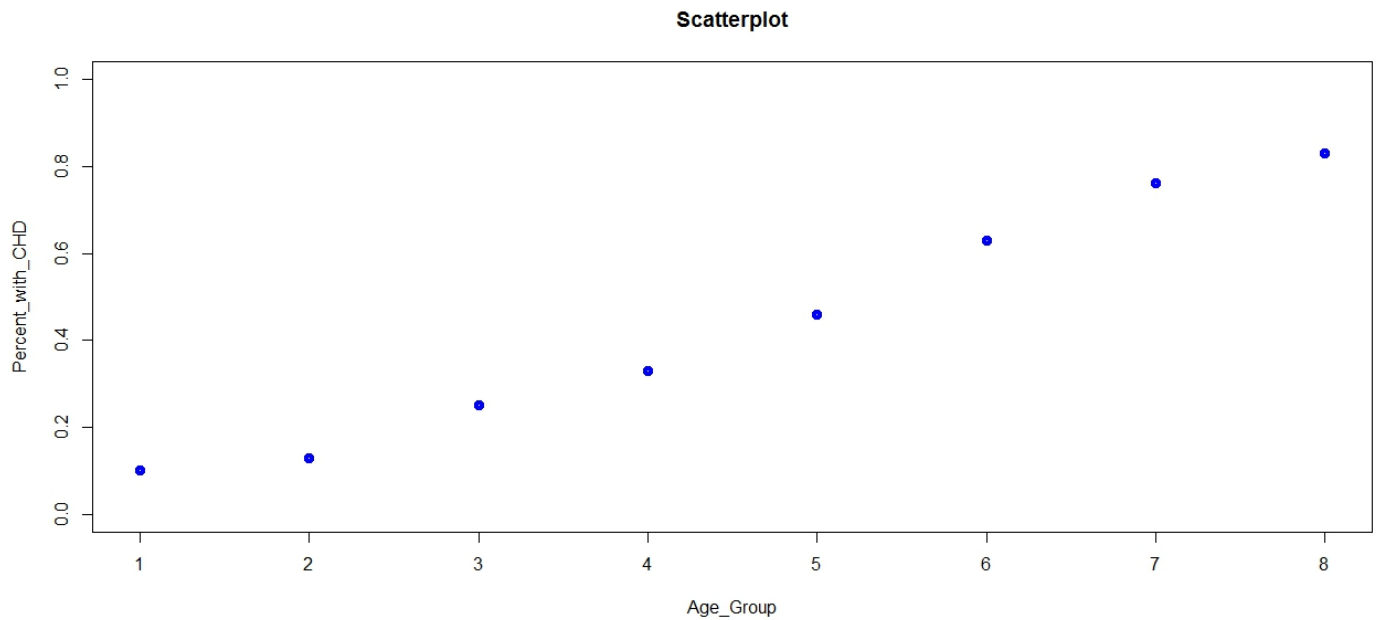
Example: Developing a Logistic Regression Model

- To clarify the relationship, we use the AGRP and compute the Average frequence of CHD per age category:

Age Group	AGRP	Absent	Present	Mean Proportion within Age Group
20-29	1	9	1	10%
30-34	2	13	2	13%
35-39	3	9	3	25%
40-44	4	10	5	33%
45-49	5	7	6	46%
50-54	6	3	5	63%
55-59	7	4	13	76%
60-69	8	2	8	83%

Example: Developing a Logistic Regression Model

- Step 1: Scatterplot The structure of the relationship between CHD and Age Group is now clear.



Example: Developing a Logistic Regression Model

- What we have done here is to compute the average number of CHD per Age Group, i.e. $E[Y|x]$.
- What we are willing to fit here is

$$E[Y|x] = \beta_0 + \beta_1 \cdot x$$

- However, since our Y variable is binary, $E[Y|x]$ will only take value between 0 and 1 with gradual increase as long as x increase.
- As a result, we need a specific function for $E[Y|x]$ reflecting these 2 properties:
  1. Values between 0 and 1
  2. Gradual increase

PART II: LOGIT TRANSFORMATION

Logit Transformation

- In order to achieve this, we use the function:

$$E[Y|x] = \pi(x) = \frac{e^{(\beta_0+\beta_1 x)}}{1 + e^{(\beta_0+\beta_1 x)}}$$

- We define the *logit transformation* in terms of $\pi(x)$ as the function $g(x)$:

$$g(x) = \ln(\frac{\pi(x)}{1-\pi(x)}) = \beta_0 + \beta_1 x$$

Logit Transformation: Difference with regression

- In the previous Chapter, we were fitting a model such as

$$y = \beta_0 + \beta_1 x + \epsilon$$

where ϵ were normally distributed.

- In our context, this does not make sense as y should take either the value 1 or the value 0.
- We should instead see it the following way

$$y = \beta_0 + \beta_1 x + \epsilon$$

where

$$\begin{array}{llll} \text{If } y = 1 & \text{then,} & \epsilon = 1 - g(x) & \text{with probability } \pi(x) \\ \text{If } y = 0 & \text{then,} & \epsilon = -g(x) & \text{with probability } 1 - \pi(x) \end{array}$$

- As a result, the errors ϵ are following a binomial distribution with probability $\pi(x)$.

Logit Transformation: Difference with regression

- As a result,

$$\begin{array}{llll} \text{If } y = 1 & \text{then,} & \epsilon = 1 - g(x) & \text{with probability } \pi(x) \\ \text{If } y = 0 & \text{then,} & \epsilon = -g(x) & \text{with probability } 1 - \pi(x) \end{array}$$

$$P(Y = y_i = 1|x_i) = \pi(x_i)^1 = \pi(x_i)^{y_i}$$

and

$$P(Y = y_i = 0|x_i) = (1 - \pi(x_i))^{1-0} = [1 - \pi(x_i)]^{1-y_i}$$

Finally,

$$P(Y = y_i|x_i) = \pi(x_i)^{y_i}[1 - \pi(x_i)]^{1-y_i}$$

PART I: MAXIMUM LIKELIHOOD

Fitting the logistic regression: MLE Method

- We recall that the maximum likelihood estimation (MLE) is a method of estimating the parameters of a statistical model given observations.
- The target is to find the parameter values that maximize the likelihood of making the observations given the parameters.
- The likelihood function is mathematically defined as

$$l(\beta) = \prod_{i=1}^{n} f(y_i|x_i)$$

- We are willing to maximize this likelihood function.

Fitting the logistic regression: MLE Method

In our case

- β is the vector of parameters we are looking for, i.e. $\beta = (\beta_0, \beta_1)$.
- In our case, we have already found the contribution for each data $f(y_i|x_i)$, so that

$$\begin{aligned} l(\beta) &= \prod_{i=1}^{n} f(y_i|x_i) \\ &= \prod_{i=1}^{n} \pi(x_i)^{y_i}[1-\pi(x_i)]^{1-y_i} \end{aligned}$$

- In order to maximize we have to work with the derivatives with respect to β_0 and β_1.

Fitting the logistic regression: MLE Method

Maximizing the likelihood is equivalent to maximizing the log-likelihood, so that we consider $L(\beta)$

$$\begin{aligned}
L(\beta) &= \log(l(\beta)) \\
&= \log(\prod_{i=1}^{n} \pi(x_i)^{y_i}[1-\pi(x_i)]^{1-y_i}) \\
&= \sum_{i=1}^{n} [y_i \cdot \log(\pi(x_i)) + (1-y_i) \cdot \log(1-\pi(x_i))] \\
&= \sum_{i=1}^{n} [\log(1-\pi(x_i)) + y_i \cdot \log(\frac{\pi(x_i)}{1-\pi(x_i)})] \\
&= \sum_{i=1}^{n} [-\log(1+e^{\beta_0+\beta_1 x_i}) + y_i \cdot (\beta_0+\beta_1 x_i)]
\end{aligned}$$

MLE Method: First parameter

- Computing the first derivative with respect to the first parameter β_0

$$\begin{aligned}
\frac{\partial L(\beta)}{\partial \beta_0} &= \frac{\partial}{\partial \beta_0} \sum_{i=1}^{n} [-\log(1 + e^{\beta_0 + \beta_1 x_i}) + y_i(\beta_0 + \beta_1 x_i)] \\
&= \sum_{i=1}^{n} [-\frac{1}{1 + e^{\beta_0 + \beta_1 x_i}}(e^{\beta_0 + \beta_1 x_i}) + y_i] \\
&= \sum_{i=1}^{n} [-\pi(x_i) + y_i]
\end{aligned}$$

MLE Method: First parameter

- The first derivative leads to

$$\sum_{i=1}^{n} [-\pi(x_i) + y_i] = 0 \Leftrightarrow \sum \widehat{\pi(x_i)} = \sum y_i$$

- However, it is not possible to go further as the equation is not linear.
- We thus need some computer algorithms to solve this problem.
- The equation means that the sum of our predicted values are equal to the sum of our response data.

MLE Method: Second parameter

- Computing the first derivative with respect to the second parameter β_1

$$\begin{aligned}
\frac{\partial L(\beta)}{\partial \beta_1} &= \frac{\partial}{\partial \beta_1} \sum_{i=1}^{n} [-\log(1 + e^{\beta_0 + \beta_1 x_i}) + y_i(\beta_0 + \beta_1 x_i)] \\
&= \sum_{i=1}^{n} [-x_i \frac{1}{1 + e^{\beta_0 + \beta_1 x_i}} (e^{\beta_0 + \beta_1 x_i}) + y_i \cdot x_i] \\
&= \sum_{i=1}^{n} [x_i \cdot (-\pi(x_i) + y_i)]
\end{aligned}$$

MLE Method: Second parameter

- The other derivative leads to

$$\sum_{i=1}^{n} x_i[-\pi(x_i) + y_i] = 0$$

- However, it is not possible to go further as the equation is again not linear.

MLE Method: Conclusions

- A direct algebric equation is not possible when fitting with the MLE method.
- The MLE method expressed in the one dimension case (only one dependant variable x) is easily generalized to multiple cases.

PART III: MODEL RESULTS

Example: Model Outcome

- We get the following outcome for our model

```
> mymodel <- glm(CHD ~ Age,family=binomial (link='logit'))
> summary(mymodel)

Call:
glm(formula = CHD ~ Age, family = binomial(link = "logit"))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9702  -0.8447  -0.4572   0.8264   2.2866

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -5.31026    1.13463  -4.680 2.87e-06 ***
Age          0.11088    0.02407   4.606 4.11e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 136.66  on 99  degrees of freedom
Residual deviance: 107.40  on 98  degrees of freedom
AIC: 111.4

Number of Fisher Scoring iterations: 4
```

Predicting a value and logit function

- The $\widehat{\pi(x_i)}$ is going to give us the model prediction

$$\begin{aligned}\widehat{\pi(x_i)} &= \frac{e^{\hat{\beta}_0+\hat{\beta}_1\cdot x_i}}{1+e^{\hat{\beta}_0+\hat{\beta}_1\cdot x_i}} \\ &= \frac{e^{-5.31+0.11\cdot x_i}}{1+e^{-5.31+0.11\cdot x_i}}\end{aligned}$$

For somebody aged 43 years, one gets

$$\begin{aligned}\widehat{\pi(x_i)} &= \frac{e^{-5.31+0.11\cdot 43}}{1+e^{-5.31+0.11\cdot 43}} \\ &= 0.367\end{aligned}$$

- Finally, our logit function is given by

$$\widehat{g(x_i)} = -5.31 + 0.11 \cdot x_i = -0.58$$

Prediction interpretation

- For somebody aged 43 years, one gets

$$\begin{aligned} \widehat{\pi(x_i)} &= 0.367 \\ \widehat{g(x_i)} &= -0.58 \end{aligned}$$

- The interpretation of $g(x_i) = -0.58$ is not clear. It is similar to a score.
- The interpretation of $\widehat{\pi(x_i)} = 0.367$ is easier because $\widehat{\pi(x_i)}$ is a probability.
- A low score for $\widehat{\pi(x_i)}$ will tend to predict a 0 value for our response variable $\widehat{y_{43}}$.
- A high score for $\widehat{\pi(x_i)}$ will tend to predict a 1 for $\widehat{y_i}$.
- The threshold level (cutting point) will have to be decided: at which level of $\widehat{\pi(x_i)}$ do we consider that we predict a 0 or a 1.
- Generally, we decide with the 0.5 level.

PART IV: LIKELIHOOD RATIO TEST

Testing for the significance of the model

- The idea is the same as in the previous Chapter. We estimate the model quality by comparing our observed data y_i to predicted data $\widehat{y_i}$
- In our context of likelihood, the tool we are going to use is called the **likelihood ratio** D:

$$\begin{aligned} D &= -2 \cdot \log\left[\frac{\text{likelihood of a 1st model}}{\text{likelihood of a 2nd model}}\right] \\ &= -2 \cdot (\log[\text{likelihood of a 1st model}] \\ &\quad - \log[\text{likelihood of a 2nd model}]) \end{aligned}$$

- The main use of Likelihood ratio is to establish models comparison by comparing their likelihood.
- As a result D is often called the deviance from one model to the other.

Testing for the significance of the model

- The deviance D plays the same role as the SSE is the multiple linear regression context. It is a measure of goodness of fit.
- The value -2 at the beginning of the equation is for fitting with the probability distribution.
- For instance, we can compare any model to the perfect model "**saturated model**".
- A perfect model will exactly represent each y_i. This **saturated model** has as many parameters as numbers of data and "makes no mistake".
- In the context of a saturated model, we make no mistake and $\pi(x_i) = y_i$.

Testing for the significance of the model

- Since y_i takes values either 0 or 1, we get for a saturated model

$$\begin{aligned} l(\beta) &= \prod_{i=1}^{n} \pi(x_i)^{y_i}[1-\pi(x_i)]^{1-y_i} \\ &= \prod_{i=1}^{n} y_i^{y_i}[1-y_i)]^{1-y_i} \\ &= 1 \end{aligned}$$

- Our likelihood ratio is reduced to

$$D = -2 \cdot \log[\text{likelihood of the fitted model}]$$

Comparing Model

- The idea is to find which models offer a better fit.
- Of course, the more we have predictors, the best fit we will get.
- However, the improvements may not be that significant regarding the increase in model complexity.
- Suppose we compare 2 models: Model 1 with more complexity/variables (Alternative Model), Model 2 with fewer variables (Null Model).

$$\begin{aligned} H_0 &:= \text{ the 2 models are equivalent, we choose Model 2} \\ H_1 &:= \text{ Model 1 offers a better fit} \end{aligned}$$

Comparing Model

- We compare 2 models: Model 1 with more complexity/variables, Model 2 with fewer variables.

$$\begin{aligned} H_0 &:= \text{ the 2 models are equivalent, we choose Model 2} \\ H_1 &:= \text{ Model 1 offers a better fit} \end{aligned}$$

- We compute

$$\begin{aligned} D &= -2 \cdot \log\left[\frac{\text{likelihood of Model 2 (Null Model)}}{\text{likelihood of Model 1 (Alternative Model)}}\right] \\ &= +2 \cdot \log\left[\frac{\text{likelihood of Model 1}}{\text{likelihood of Model 2}}\right] \\ &= -2 \cdot (\log[\text{likelihood of Model 2}] \\ &\quad - \log[\text{likelihood of Model 1}]) \end{aligned}$$

- If the difference is significant (p-value criteria), then the model improvement with increasing complexity is of interest and we choose Model 1.

Comparing Model

- We use the German Credit data (62 characteristics and 1000 observations).
- The target is to estimate if a client is a bad consumer (0) or a good consumer (1) according to several characteristics.

```
> ########################Loading Data
>
> library(caret)
>
> data(GermanCredit)
>
> ######Defining Complex model: Model 1
> mod_fit_one <- glm(Class ~ Age + ForeignWorker + Property.RealEstate + Housing.Own +
+                      CreditHistory.Critical, data=GermanCredit, family="binomial")
>
> ######Defining Simpler model: Model 2
> mod_fit_two <- glm(Class ~ Age + ForeignWorker, data=GermanCredit, family="binomial")
>
> library(lmtest)
> lrtest(mod_fit_one, mod_fit_two)
Likelihood ratio test

Model 1: Class ~ Age + ForeignWorker + Property.RealEstate + Housing.Own +
    CreditHistory.Critical
Model 2: Class ~ Age + ForeignWorker
  #Df  LogLik Df  Chisq Pr(>Chisq)
1   6 -575.42
2   3 -602.48 -3 54.124  1.056e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Complex model is significantly better.

Pseudo-R Squared

- In the context of logistic regression, there is no R^2, meaning no analysis of the part of the response variable explained.
- We will work with Pseudo R^2 for which the most famous is the McFadden.
- The McFadden is defined by

$$R^2_{McF} = 1 - \frac{\log(LM)}{\log(L_0)}$$

where $log(LM)$ is the log likelihood value for the fitted model and $log(L_0)$ is the log likelihood for the null (constant) model with only an intercept as a predictor.

- R^2_{McF} takes values between 0 and 1. Small values (close to 0) present models with few predicting capacity.

Pseudo-R Squared

- In our example, one gets

```
> library(pscl)
> pR2(mod_fit_one)
         llh       llhNull            G2      McFadden          r2ML          r2CU
-575.41710690 -610.86430205   70.89439032     0.05802794    0.06843973    0.09703913
```

- The level of explanation is quite poor. As previously anticipated, the situation is even worst in our second model.

```
> pR2(mod_fit_two)
         llh       llhNull            G2      McFadden          r2ML          r2CU
-602.47908713 -610.86430205   16.77042985     0.01372680    0.01663059    0.02358013
```

Overall Prediction Quality

- In order to evaluate the model quality, we simply compare the prediction with the realized data.
- To prevent from overfitting, we usually, start by working on sample data and test the prediction ability on new data.
- After partitioning the data and training the model, we finally get our model predicting quality estimate

```
> Train <- createDataPartition(GermanCredit$Class, p=0.6, list=FALSE)
> training <- GermanCredit[ Train, ]
> testing <- GermanCredit[ -Train, ]
>
> mod_fit <- train(Class ~ Age + ForeignWorker + Property.RealEstate + Housing.Own + CreditHistory.Critical,  da
>
> pred = predict(mod_fit, newdata=testing)
> accuracy <- table(pred, testing[,"Class"])
> sum(diag(accuracy))/sum(accuracy)
[1] 0.7025
> accuracy

pred   Bad Good
  Bad   24   23
  Good  96  257
```

Individual Prediction Quality

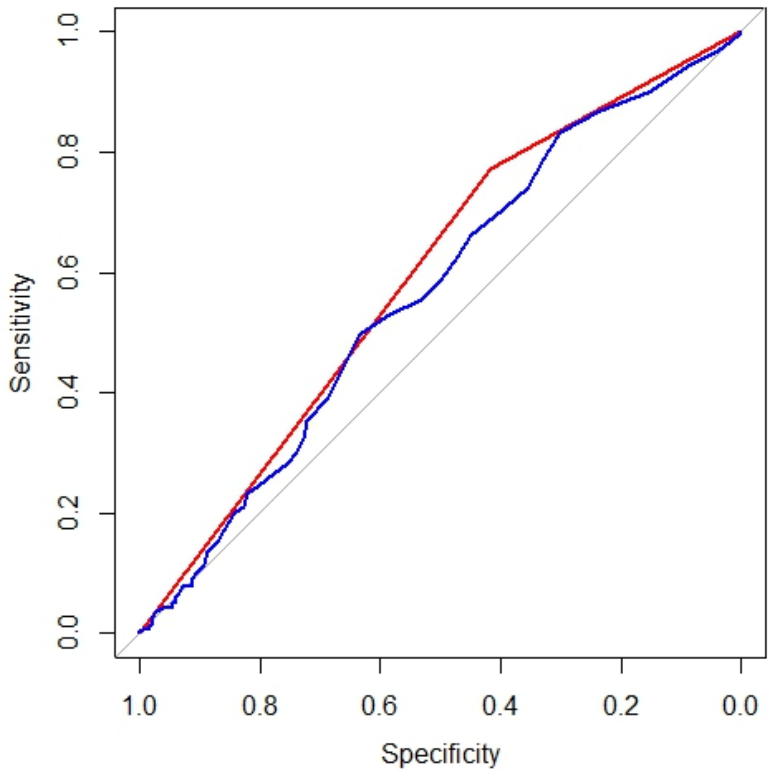
- After looking at the model ability to predict correctly the response variable y, one might be willing to look at the independant variable individually.
- To achieve this task, a useful tool is the ROC curve. To introduce the ROC curve, we need to define the **sensitivity**, the **specificity** and the **cut-point**.
- The **sensitivity** is the proportion of truly positive observations which is classified as such by the model or test (indicating $y = 1$ correctly).
- Conversely the **specificity** is the probability of the model predicting 'negative' given that the observation is 'negative' (indicating $y = 0$ correctly).

Individual Prediction Quality

- To decide, if we are going to indicate $y = 0$ or $y = 1$, we need a decision rule, meaning a **cut-point c**. Generally speaking, the cut-off point is 0.5.
- We classify the observations with a fitted probability above c as positive and those at or below it as negative.
- For this particular cut-off, we can estimate the sensitivity by the proportion of observations with $Y = 1$ which have a predicted probability above c, and similarly we can estimate specificity by the proportion of $Y = 0$ observations with a predicted probability at or below c.
- There is always a trade-off between **specificity** and **sensitivity** because if one increases, the other decreases.

Individual Prediction Quality

- We come to the ROC curve, which is simply a plot of the values of sensitivity against one minus specificity, as the value of the cut-point c is increased from 0 through to 1.
- We can apply this to either the full model or individual variable.



Individual Prediction Quality

- A perfect model would predict perfectly the (**sensitivity**), i.e. reaching 100% of correct answer. At the same time, it would make no mistake for predicting negative answers (1-**specificity**).
- As a result, a perfect model would reach the upper left corner of our ROC graph.
- It means that we can estimate the model quality by computing the area of the ROC curve above the purely random model represented by the straight line.
- If this area is high, we have a good discrimination level.
- The area takes value between 0.5 and 1. A value above 0.8 would be considered as a good level.

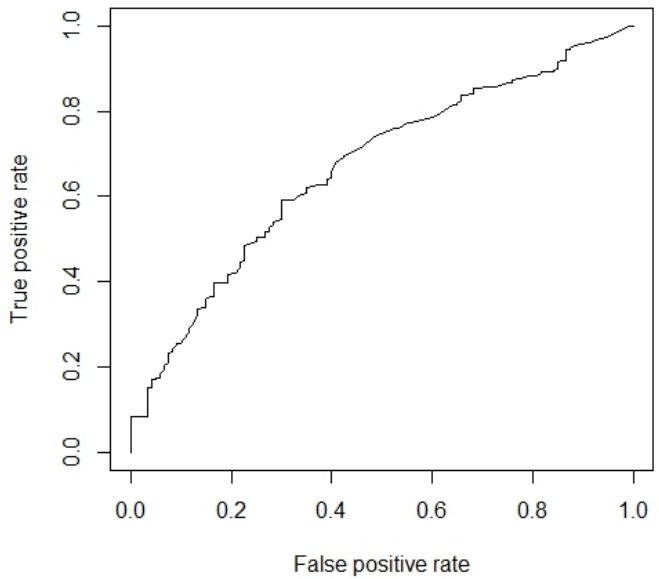
Individual Prediction Quality

- Individual application with in red the Housing Owned curve and in blue the Age one.

```
> library(pROC)
> # Compute AUC for predicting Class with the variable CreditHistory.Critical
> f1 = roc(Class ~ Housing.Own, data=training)
> f2 = roc(Class ~ Age, data=training)
> f1

Call:
roc.formula(formula = Class ~ Housing.Own, data = training)

Data: Housing.Own in 180 controls (Class Bad) < 420 cases (Class Good).
Area under the curve: 0.594
> f2

Call:
roc.formula(formula = Class ~ Age, data = training)

Data: Age in 180 controls (Class Bad) < 420 cases (Class Good).
Area under the curve: 0.5684
```

Individual Prediction Quality

- We can develop the same tool for the overall model



Individual Prediction Quality

- And we get the following area

```
> library(ROCR)
> # Compute AUC for predicting Class with the model
> prob <- predict(mod_fit_one, newdata=testing, type="response")
> pred <- prediction(prob, testing$Class)
> perf <- performance(pred, measure = "tpr", x.measure = "fpr")
> plot(perf)
> auc <- performance(pred, measure = "auc")
> auc <- auc@y.values[[1]]
> auc
[1] 0.6676339
```

Executive Summary

- Logistic Regression specificities: 0 and 1.
- Model fitting through MLE method.
- No diagnostic plot: qqplot does not make sense.
- Model outcome: the score function and interpretation
- Model comparison and selection: the likelihood ratio test
- Prediction quality analysis: Pseudo R^2, Overall prediction, ROC curve.