Logistic regression: introduction

Final data structure

```
> str(training_set)

'data.frame': 19394 obs. of  8 variables:
 $ loan_status   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ loan_amnt     : int  25000 16000 8500 9800 3600 6600 3000 7500 6000 22750 ...
 $ grade         : Factor w/ 7 levels "A","B","C","D",..: 2 4 1 2 1 1 1 2 1 1 ...
 $ home_ownership: Factor w/ 4 levels "MORTGAGE","OTHER",..: 4 4 1 1 1 3 4 3 4 1 ...
 $ annual_inc    : num  91000 45000 110000 102000 40000 ...
 $ age           : int  34 25 29 24 59 35 24 24 26 25 ...
 $ emp_cat       : Factor w/ 5 levels "0-15","15-30",..: 1 1 1 1 1 2 1 1 1 1 ...
 $ ir_cat        : Factor w/ 5 levels "0-8","11-13.5",..: 2 3 1 4 1 1 1 4 1 1 ...
```

What is logistic regression?

A regression model with output between 0 and 1

$$P(\text{loan\_status} = 1 \mid x_1, \ldots, x_m) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m)}}$$

$x_1, \ldots, x_m$ →

```
loan_amnt     grade     age     annual_inc
home_ownership     emp_cat     ir_cat
```

$\beta_0, \ldots, \beta_m$ → Parameters to be estimated

$\beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m$ → Linear predictor

Fitting a logistic model in R

$\hat{\beta}_0$

$\hat{\beta}_1$

```
> log_model <- glm(loan_status ~ age , family= "binomial", data = training_set)
> log_model

Call:  glm(formula = loan_status ~ age, family = "binomial", data = training_set)

Coefficients:
(Intercept)          age
 -1.793566    -0.009726

Degrees of Freedom: 19393 Total (i.e. Null);  19392 Residual
Null Deviance:     13680
Residual Deviance: 13670  AIC: 13670
```

$$P(\text{loan\_status} = 1 \mid \text{age}) = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 age)}}$$

Probabilities of default

$$P(\text{loan\_status} = 1 \mid x_1, \ldots, x_m) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m)}} = \frac{e^{\beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m}}{1 + e^{\beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m}}$$

$$P(\text{loan\_status} = 0 \mid x_1, \ldots, x_m) = 1 - \frac{e^{\beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m}}{1 + e^{\beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m}} = \frac{1}{1 + e^{\beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m}}$$

$$\frac{P(\text{loan\_status} = 1 \mid x_1, \ldots, x_m)}{P(\text{loan\_status} = 0 \mid x_1, \ldots, x_m)} = e^{\beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m} \longrightarrow$$ odds in favor of loan_status=1

Interpretation of coefficient

If variable x_j goes up by 1 → The odds are multiplied by e^{β_j}

$\beta_j < 0$ → $e^{\beta_j} < 1$ → The odds decrease as x_j increases

$\beta_j > 0$ → $e^{\beta_j} > 1$ → The odds increase as x_j increases

Applied to our model

If variable age goes up by 1 → The odds are multiplied by $e^{-0.009726}$

→ The odds are multiplied by 0.991

Let's practice!

Logistic regression: predicting the probability of default

An example with "age" and "home ownership"

$\hat{\beta}_0$

$\hat{\beta}_1$

$\hat{\beta}_2$

```
> log_model_small <- glm(loan_status ~ age + home_ownership, family = "binomial", data = training_set)
> log_model_small

Call:  glm(formula = loan_status ~ age + home_ownership, family = "binomial",
    data = training_set)

Coefficients:
 (Intercept)              age     home_ownershipOTHER     home_ownershipOWN    home_ownershipRENT
 -1.886396          -0.009308           0.129776             -0.019384             0.158581

Degrees of Freedom: 19393 Total (i.e. Null);  19389 Residual
Null Deviance:      13680
Residual Deviance: 13660   AIC: 13670
```

$\hat{\beta}_4$

$\hat{\beta}_3$

$$P(\text{loan\_status} = 1 \mid \text{age}, \text{home\_ownership}) = \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 \text{age} + \hat{\beta}_2 \text{OTHER} + \hat{\beta}_3 \text{OWN} + \hat{\beta}_4 \text{RENT})}}$$

Test set example

$$P(\text{loan\_status} = 1 \mid \text{age} = 33, \text{home\_ownership} = \text{RENT})$$

$$= \frac{1}{1 + e^{-(\hat{\beta}_0 + \hat{\beta}_1 * 33 + \hat{\beta}_2 * 0 + \hat{\beta}_3 * 0 + \hat{\beta}_4 * 1)}}$$

$$= \frac{1}{1 + e^{-(-1.886396 + (-0.009308) * 33 + (0.158581) * 1)}}$$

$$= 0.115579$$

Making predictions in R

```
> test_case <- as.data.frame(test_set[1,])

> test_case
  loan_status loan_amnt  grade home_ownership annual_inc  age  emp_cat   ir_cat
1       0         5000     B        RENT          24000    33    0-15     8-11

> predict(log_model_small, newdata = test_case)
        1
-2.03499
```

$$\hat{\beta}_0 + \hat{\beta}_1 \text{age} + \hat{\beta}_2 \text{OTHER} + \hat{\beta}_3 \text{OWN} + \hat{\beta}_4 \text{RENT}$$

```
> predict(log_model_small, newdata = test_case, type = "response")
         1
0.1155779
```

Let's practice!

Evaluating the logistic regression model result

Recap: model evaluation

```
          test_set$loan_status    model_prediction
                   …                      …
[8066,]            1                      1
[8067,]            0                      0
[8068,]            0                      0
[8069,]            0                      0
[8070,]            0                      0
[8071,]            0                      1
[8072,]            1                      0
[8073,]            1                      1
[8074,]            0                      0
[8075,]            0                      0
[8076,]            0                      0
[8077,]            1                      1
[8078,]            0                      0
[8079,]            0                      1
                   …                      …
```

model prediction

actual loan status

	no default (0)	default (1)
no default (0)	**8**	**2**
default (1)	**1**	**3**

In reality...

```
          test_set$loan_status    model_prediction
                     …                    …
[8066,]              1               0.09881492
[8067,]              0               0.09497852
[8068,]              0               0.21071984
[8069,]              0               0.04252119
[8070,]              0               0.21110838
[8071,]              0               0.08668856
[8072,]              1               0.11319341
[8073,]              1               0.16662207
[8074,]              0               0.15299176
[8075,]              0               0.08558058
[8076,]              0               0.08280463
[8077,]              1               0.11271048
[8078,]              0               0.08987446
[8079,]              0               0.08561631
                     …                    …
```

model prediction

actual loan status

	no default (0)	default (1)
no default (0)	?	?
default (1)	?	?

In reality...

```
          test_set$loan_status   model_prediction
                    …                   …
[8066,]             1              0.09881492
[8067,]             0              0.09497852
[8068,]             0              0.21071984
[8069,]             0              0.04252119
[8070,]             0              0.21110838
[8071,]             0              0.08668856
[8072,]             1              0.11319341
[8073,]             1              0.16662207
[8074,]             0              0.15299176
[8075,]             0              0.08558058
[8076,]             0              0.08280463
[8077,]             1              0.11271048
[8078,]             0              0.08987446
[8079,]             0              0.08561631
                    …                   …
```

Cutoff
or
treshold value

between 0 and 1

Cutoff = 0.5

```
          test_set$loan_status     model_prediction
                     …                      …
[8066,]              1                      0
[8067,]              0                      0
[8068,]              0                      0
[8069,]              0                      0
[8070,]              0                      0
[8071,]              0                      0
[8072,]              1                      0
[8073,]              1                      0
[8074,]              0                      0
[8075,]              0                      0
[8076,]              0                      0
[8077,]              1                      0
[8078,]              0                      0
[8079,]              0                      0
                    …                       …
```

model prediction

actual loan status	no default (0)	default (1)
no default (0)	10	0
default (1)	4	0

Accuracy = 10/(10+4+0+0) = 71.4%

Sensitivity = 0/(4+0) = 0%

Cutoff = 0.1

```
        test_set$loan_status   model_prediction
                   …                  …
[8066,]            1                  0
[8067,]            0                  0
[8068,]            0                  1
[8069,]            0                  0
[8070,]            0                  1
[8071,]            0                  0
[8072,]            1                  1
[8073,]            1                  1
[8074,]            0                  1
[8075,]            0                  0
[8076,]            0                  0
[8077,]            1                  1
[8078,]            0                  0
[8079,]            0                  0
                   …                  …
```

model prediction

actual loan status	no default (0)	default (1)
no default (0)	7	3
default (1)	1	3

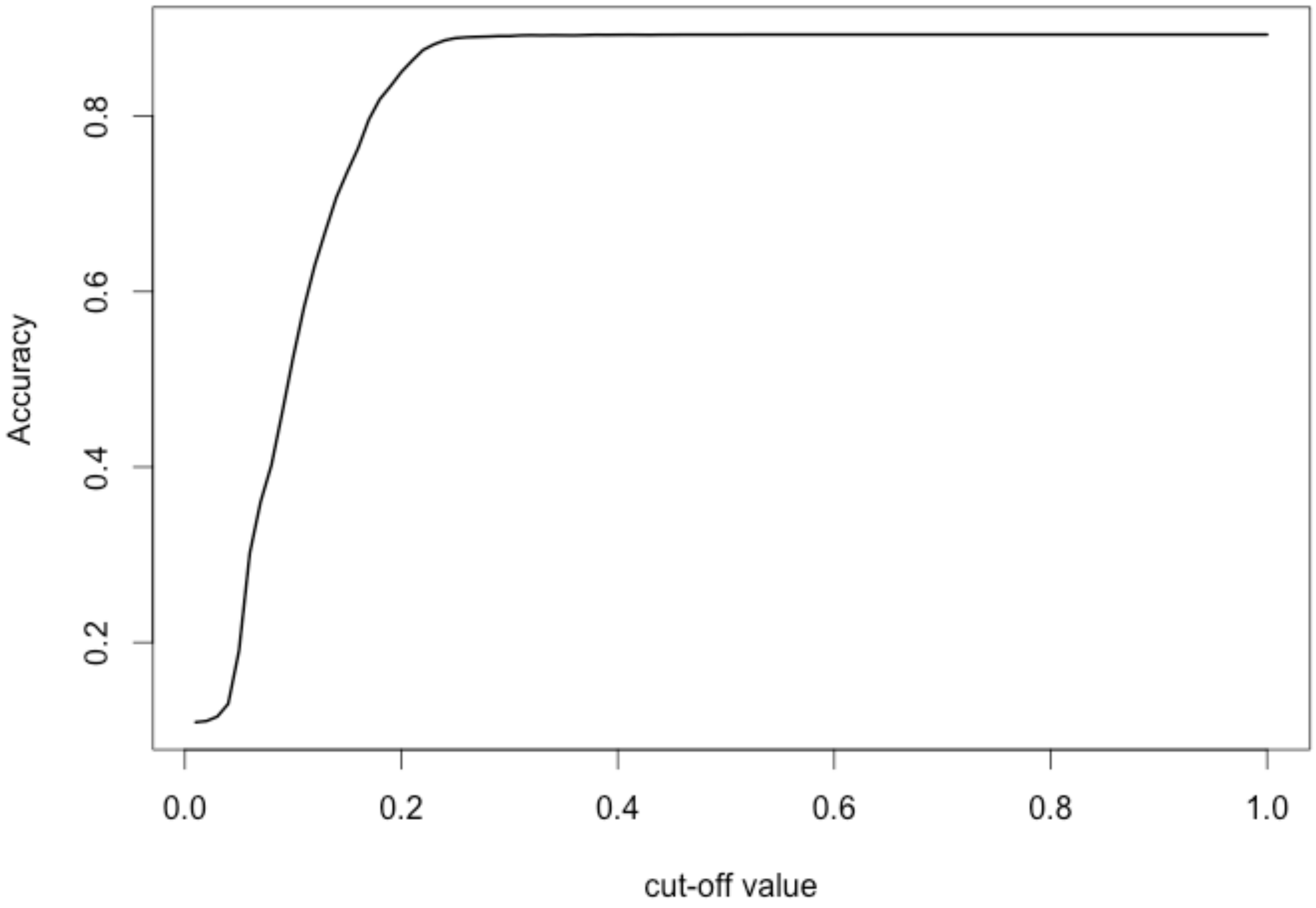
Accuracy = 10/(10+4+0+0) = 71.4%

Sensitivity = 3/(3+1) = 75%

Let’s practice!

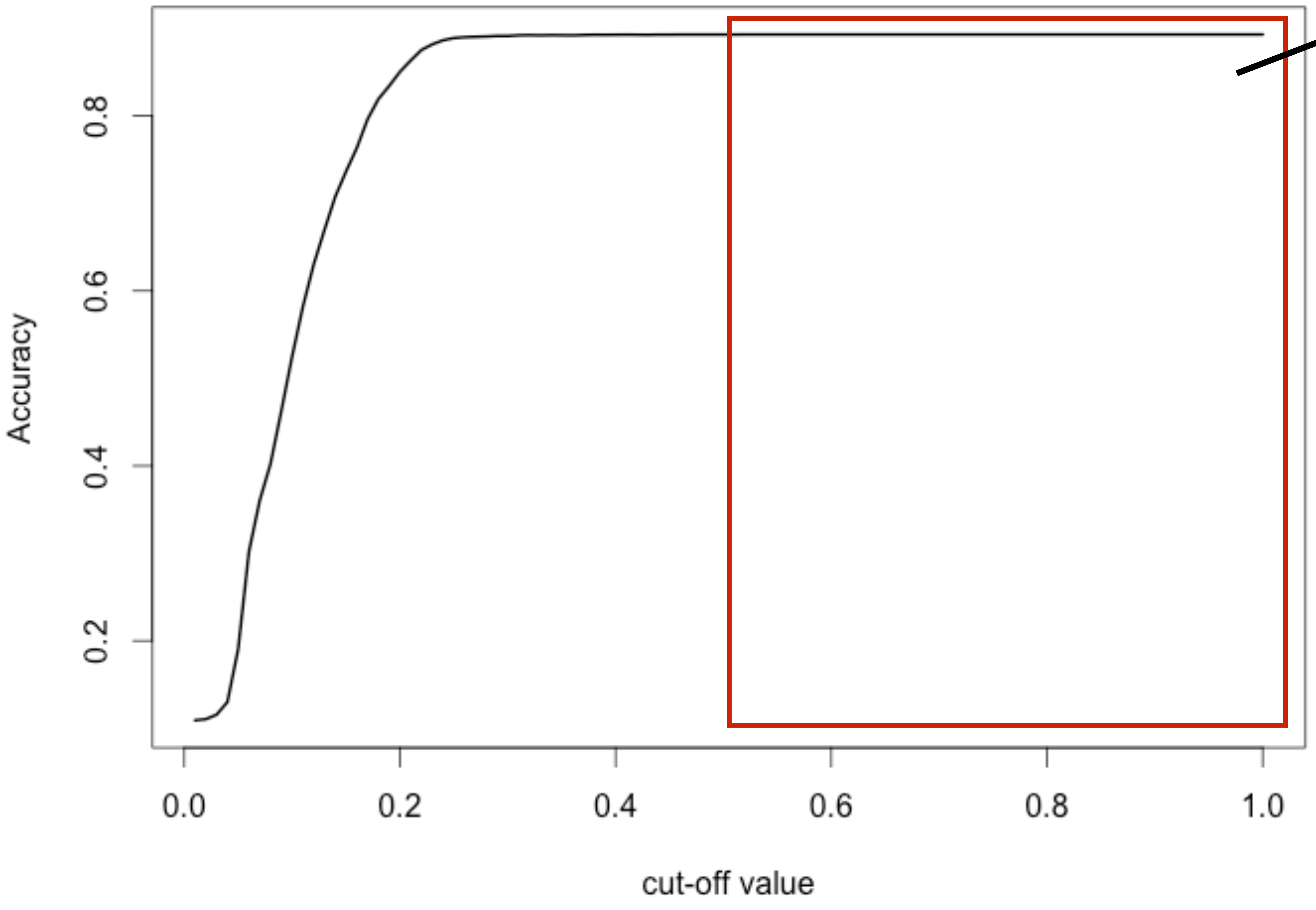
wrap-up and remarks

best cut-off for accuracy?



$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

best cut-off for accuracy?

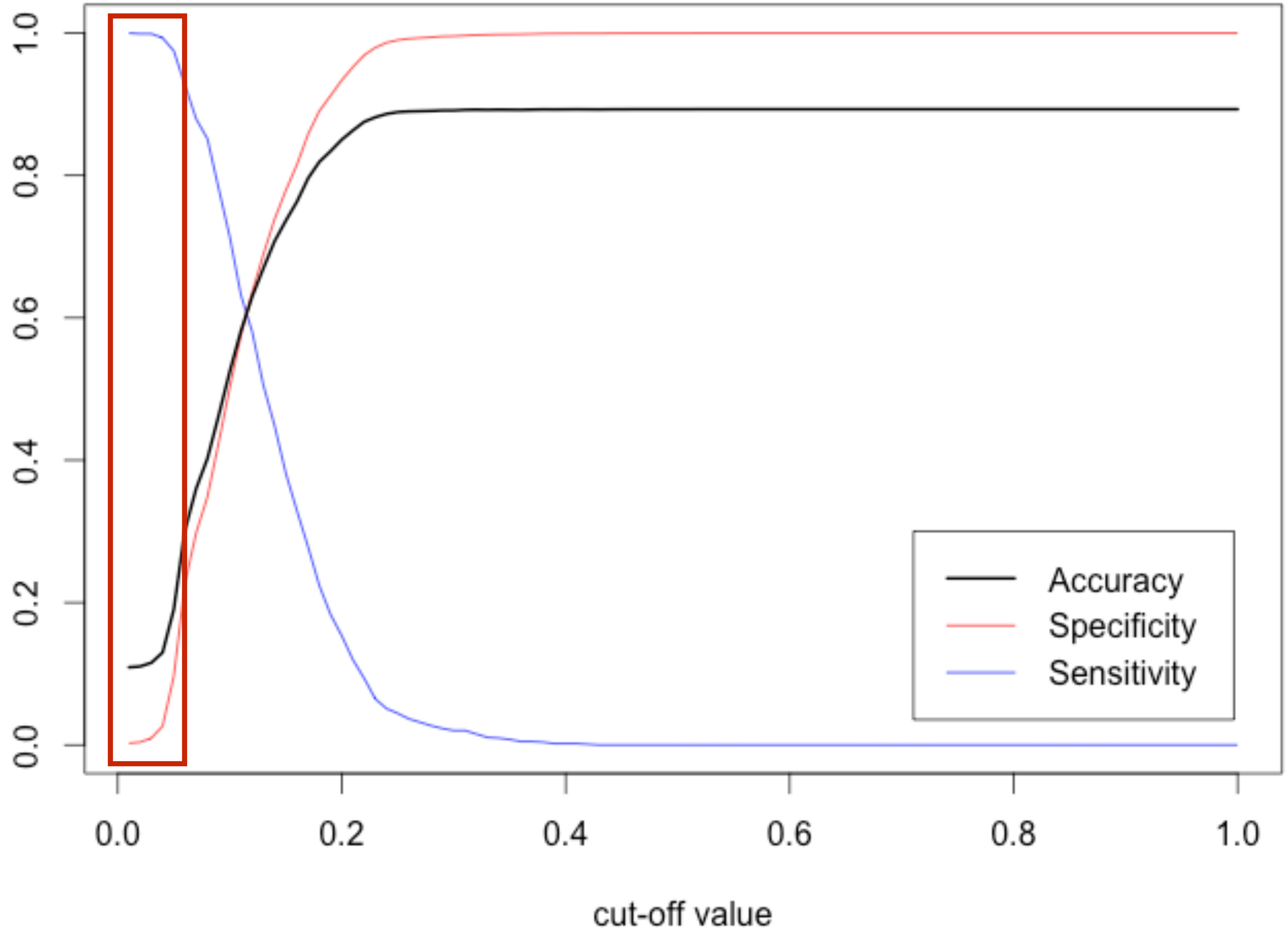


Accuracy = 89.31 %

ACTUAL defaults in test set= 10.69 % = (100 - 89.31) %

Typical result for unbalance group. High accuracy is due that for cases with >51% all cases are classified as Non-default.

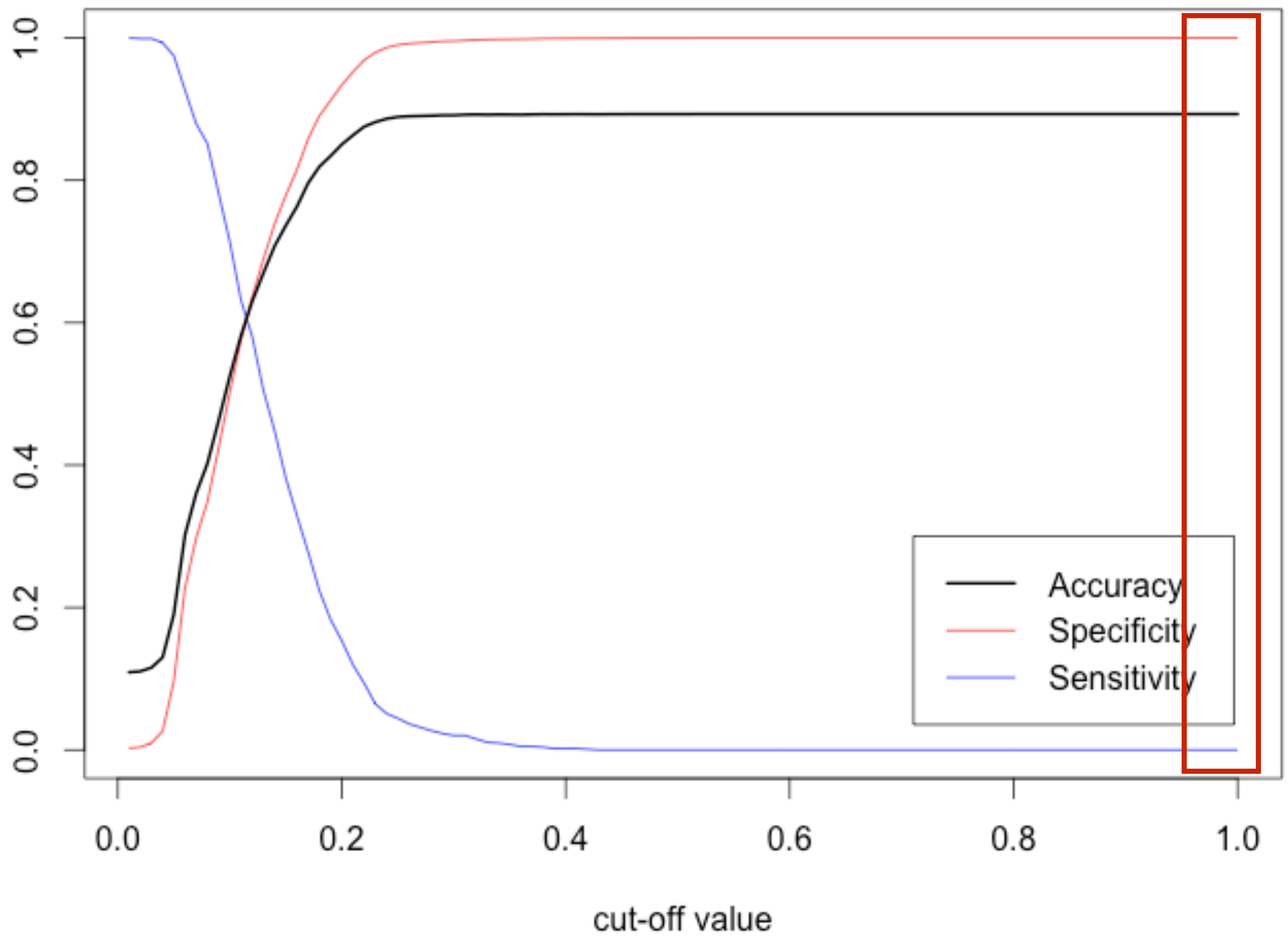
What about sensitivity or specificity?



Sensitivity = 1037 / (1037 +0) = 100%

Specificity = 0 / (0 + 864) = 0%

What about sensitivity or specificity?



Sensitivity = 0 / (0 + 1037) = 0%

Specificity = 8640 / (8640 + 0) = 100%

About logistic regression...

```
log_model_full <- glm(loan_status ~ ., family = "binomial", data = training_set)
```

is the same as

```
log_model_full <- glm(loan_status ~ ., family = binomial(link = logit),
data = training_set)
```

recall

$$P(\text{loan\_status} = 1 \mid x_1, \ldots, x_m) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m)}}$$

Other logistic regression models

```
log_model_full <- glm(loan_status ~ ., family = binomial(link = probit),
data = training_set)
```

```
log_model_full <- glm(loan_status ~ ., family = binomial(link = cloglog),
data = training_set)
```

$\beta_j < 0$ → The probability of default decreases as x_j increases

$\beta_j > 0$ → The probability of default decreases as x_j increases

BUT

$$\cancel{P(\text{loan\_status} = 1 \mid x_1, \ldots, x_m) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \ldots + \beta_m x_m)}}}$$

Let's practice!