Ch4-A Logistic Regression

Contents

4A Subsection

[ToC]

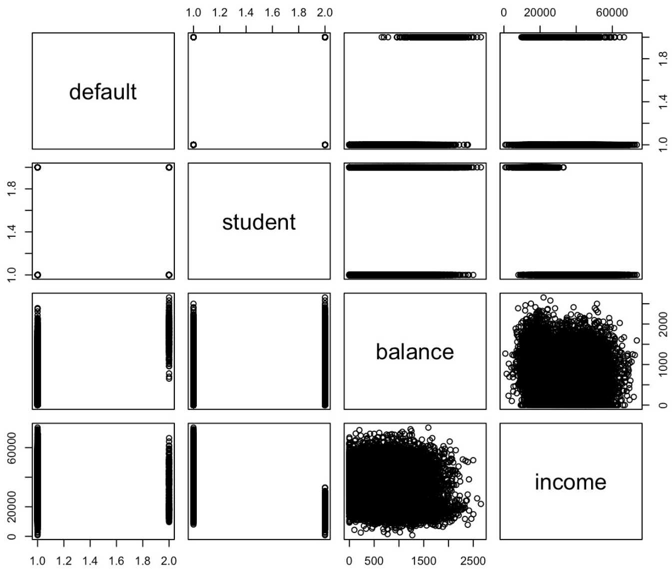
A.1 Classification Problem

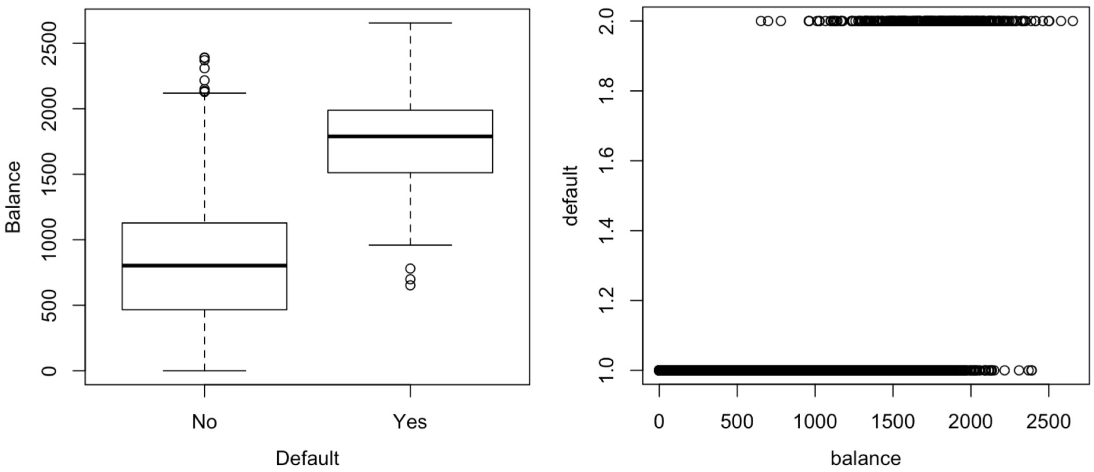
Default dataset in ISLR package:

```
library(ISLR)
data(Default)
names(Default)
  "default" "student" "balance" "income"

dim(Default)
  10000  4
```

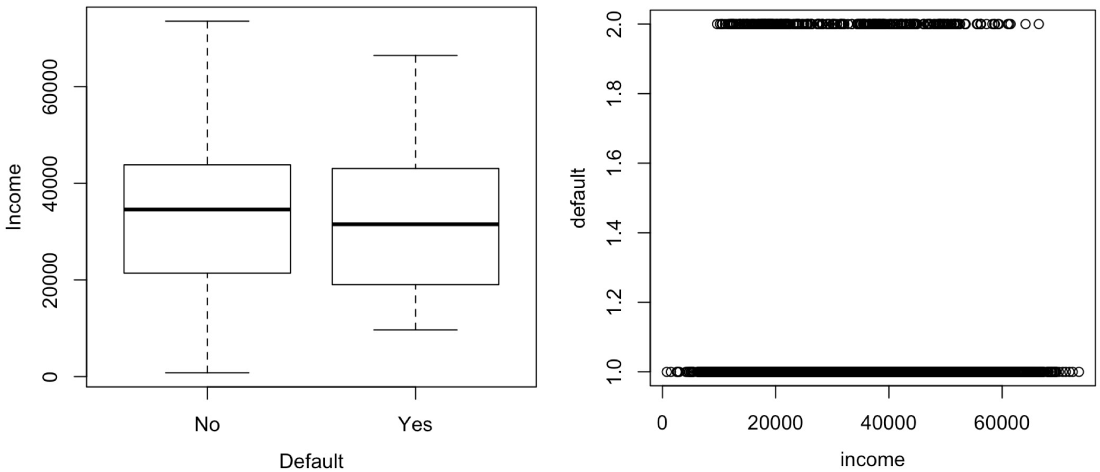
```
library(tidyverse)
Default <- as_tibble(Default)
Default
```

```
# A tibble: 10,000 x 4
   default student balance income
   <fct>   <fct>     <dbl>  <dbl>
 1 No      No         730. 44362.
 2 No      Yes        817. 12106.
 3 No      No        1074. 31767.
 4 No      No         529. 35704.
 5 No      No         786. 38463.
 6 No      Yes        920.  7492.
 7 No      No         826. 24905.
 8 No      Yes        809. 17600.
 9 No      No        1161. 37469.
10 No      No           0  29275.
# ... with 9,990 more rows
```

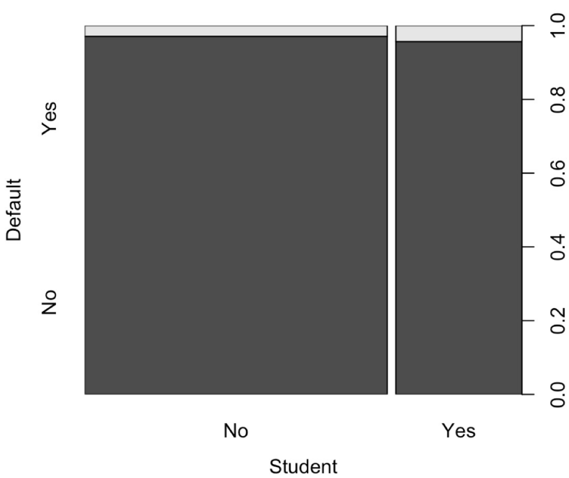

default
student
balance
income
1.0 1.2 1.4 1.6 1.8 2.0
0 20000 60000
1.0 1.4 1.8
0 1000 2000
0 20000 60000
1.0 1.2 1.4 1.6 1.8 2.0
0 500 1500 2500



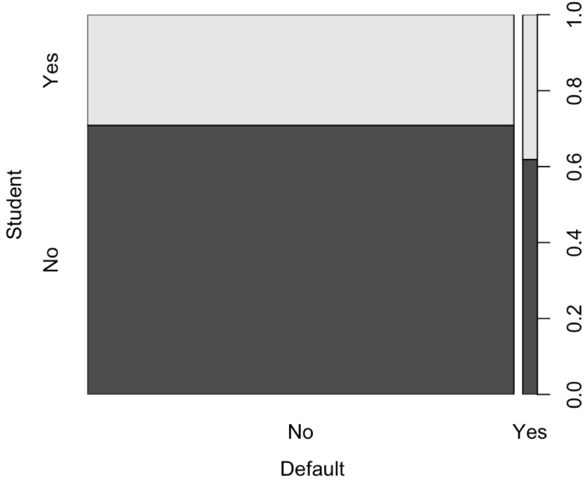
Balance
0
500
1000
1500
2000
2500
No
Yes
Default
default
1.0
1.2
1.4
1.6
1.8
2.0
0
500
1000
1500
2000
2500
balance



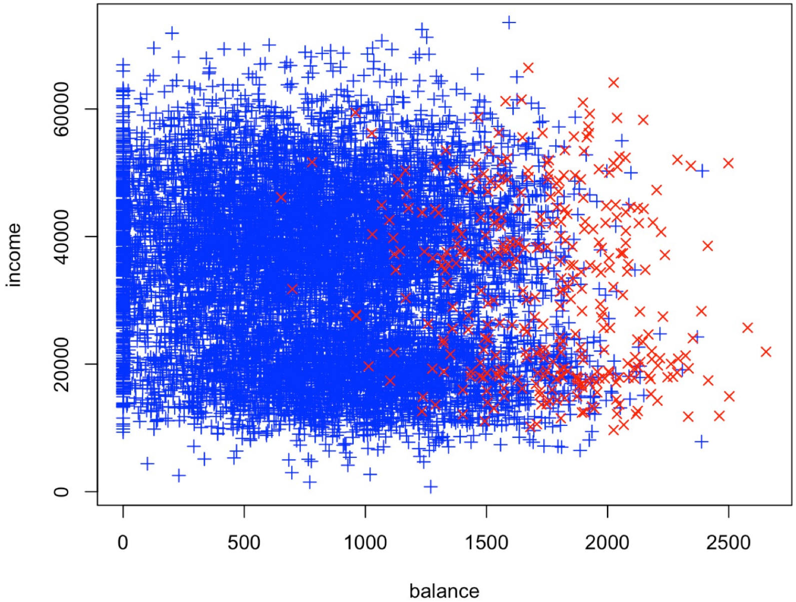
Income
0
20000
40000
60000
No
Yes
Default
default
1.0
1.2
1.4
1.6
1.8
2.0
0
20000
40000
60000
income



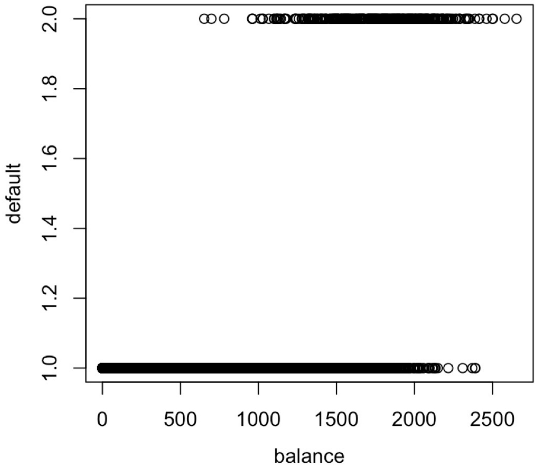
Default
Yes
No
Student
No
Yes
0.0
0.2
0.4
0.6
0.8
1.0



Student
Yes
No
Default
No
Yes
0.0
0.2
0.4
0.6
0.8
1.0


income
0
20000
40000
60000
balance
0
500
1000
1500
2000
2500

A.2 Logistic Regression



A.3 Logistic link function

- We have a binary response:

$$Y_i = \begin{cases} 1 & \text{with probability } p_i \\ 0 & \text{with probability } (1 - p_i) \end{cases}$$

$$Y_i \sim Bin(n = 1, p_i) = Bernuilli(p_i) : \quad p_i = f(X_i)$$

- Can't let p_i depend on X linearly.
- Odds has range of $[0, \inf)$

$$O_A = \frac{P(A)}{P(A')} = \frac{p_i}{1 - p_i}$$

- Let log of odds depend linearly on X,

$$\frac{p_i}{1 - p_i} = e^{\beta_0 + \beta_1 X_i}$$

- Equivalent to saying

$$\mu_i = p_i = f(X_i) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} \qquad \text{(logistic link function)}$$

- Parameters β_0 and β_1 can be estimated by Max Likelihood Method with

$$L(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{j:y_j=0} (1 - p(x_j))$$

- Can be extended to multiple predictors easily. Multiple Logistic regression:

$$p_i = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\beta_3 X_3}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\beta_3 X_3}}$$

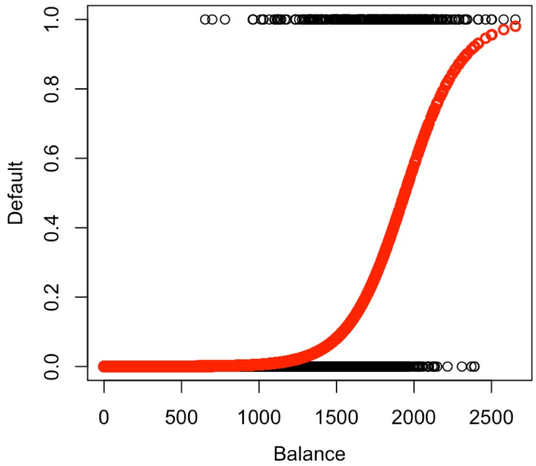
A.4 Fitting Default Data

Training : 90000 Testing: 10000 seed: 8346

Using just `balance`

```
Fit1 <- glm(default ~ balance, family=binomial, data=Train.set)
summary(Fit1)

        Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
        (Intercept) -1.066e+01  3.821e-01  -27.90   <2e-16 ***
        balance      5.493e-03  2.331e-04   23.56   <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: 2590.1  on 8999  degrees of freedom
Residual deviance: 1419.7  on 8998  degrees of freedom
AIC: 1423.7
```



A.5 Using all three predictors

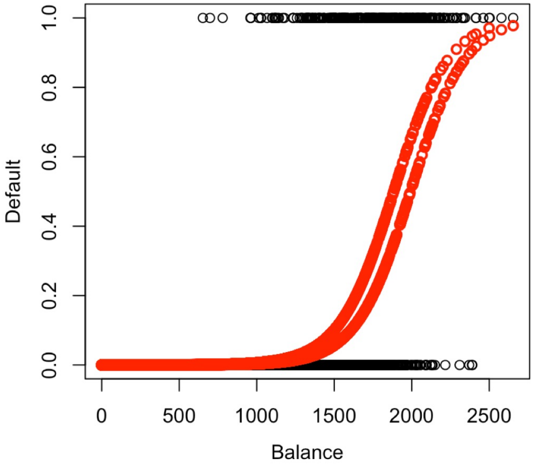
```
Fit3 <- glm(default~balance+income+student, family=binomial, data=Train.set))
summary(Fit3)

        Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
        (Intercept) -1.080e+01  5.211e-01 -20.732   <2e-16 ***
        studentYes  -5.872e-01  2.502e-01  -2.347   0.0189 *
        balance      5.706e-03  2.448e-04  23.306   <2e-16 ***
        income       1.358e-06  8.772e-06   0.155   0.8770
        ---
        Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

            Null deviance: 2590.1  on 8999  degrees of freedom
        Residual deviance: 1403.0  on 8996  degrees of freedom
```

AIC: 1411

A.6 Dropping *income* from above

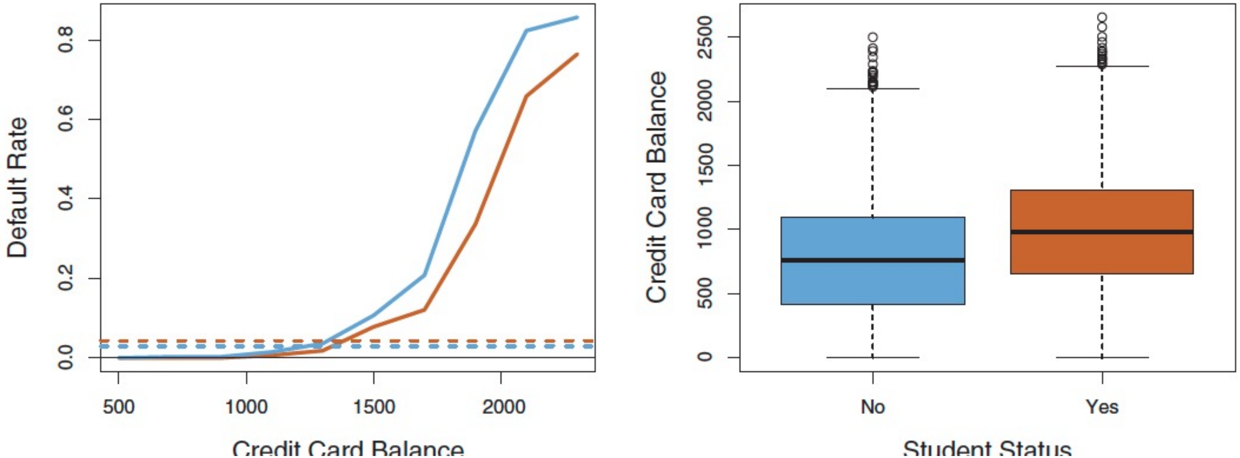
```
Fit4 <- glm(default~balance+student, family=binomial, data=Train.set)
summary(Fit4)

        Coefficients:
                      Estimate Std. Error z value Pr(>|z|)
        (Intercept) -1.075e+01  3.896e-01 -27.593  < 2e-16 ***
        balance      5.707e-03  2.448e-04  23.313  < 2e-16 ***
        studentYes  -6.176e-01  1.548e-01  -3.988 6.65e-05 ***
        ---
        Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


            Null deviance: 2590.1  on 8999  degrees of freedom
        Residual deviance: 1403.1  on 8997  degrees of freedom
```

```
AIC: 1409.1
```



A.7 Being a Student

coefficient for `student` predictor is negative, and significant.



A.8 Confusion Matrix

```
library(caret)  # for confusionMatrix

#- Pick a threshold
threshold = .5

#- Check the training set accuracy
library(caret)
Train.pred = ifelse(Train.prob > threshold, "Yes", "No")  # Turn the fitted values to Up/Down
Test.pred  = ifelse(Test.prob  > threshold, "Yes", "No")
CM.train <- confusionMatrix(factor(Train.pred), factor(as.matrix(Train.resp)), positive="Yes")
CM.test  <- confusionMatrix(factor(Test.pred),  factor(as.matrix(Test.resp)), positive="Yes")
```

```
> CM.train             # Training set result
Confusion Matrix and Statistics

          Reference
Prediction   No  Yes
       No  8672  203
       Yes   34   91

               Accuracy : 0.9737
                 95% CI : (0.9701, 0.9769)
    No Information Rate : 0.9673
    P-Value [Acc > NIR] : 0.0002764

                  Kappa : 0.4231

 Mcnemar's Test P-Value : < 2.2e-16
```

```
            Sensitivity : 0.30952
            Specificity : 0.99609
         Pos Pred Value : 0.72800
         Neg Pred Value : 0.97713
             Prevalence : 0.03267
         Detection Rate : 0.01011
   Detection Prevalence : 0.01389
      Balanced Accuracy : 0.65281

       'Positive' Class : Yes
```

```
> CM.test             # Testing set
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  957  25
       Yes   4  14

               Accuracy : 0.971
                 95% CI : (0.9586, 0.9805)
    No Information Rate : 0.961
    P-Value [Acc > NIR] : 0.0555850

                  Kappa : 0.4784

 Mcnemar's Test P-Value : 0.0002041
```

```
         Sensitivity : 0.3590
         Specificity : 0.9958
      Pos Pred Value : 0.7778
      Neg Pred Value : 0.9745
          Prevalence : 0.0390
      Detection Rate : 0.0140
Detection Prevalence : 0.0180
   Balanced Accuracy : 0.6774

    'Positive' Class : Yes
```

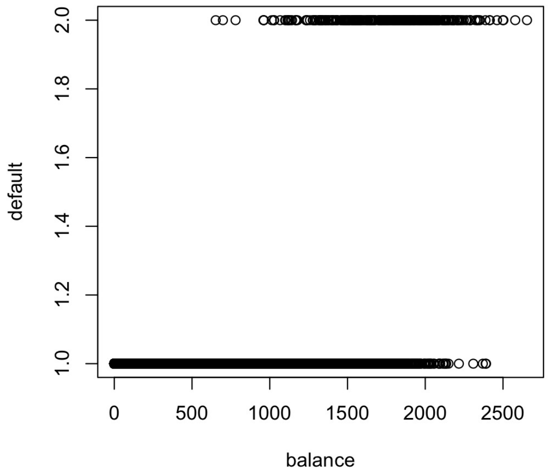
A.9 Confusion Matrix

```
          Reference
Prediction  No Yes
       No  957  25
       Yes   4  14
```

		Predicted class		
		− or Null	+ or Non-null	Total
True	− or Null	True Neg. (TN)	False Pos. (FP)	N
class	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N*	P*	

Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1−Specificity
True Pos. rate	TP/P	1−Type II error, power, sensitivity, recall
Pos. Pred. value	TP/P*	Precision, 1−false discovery proportion
Neg. Pred. value	TN/N*	

A.10 Threshold



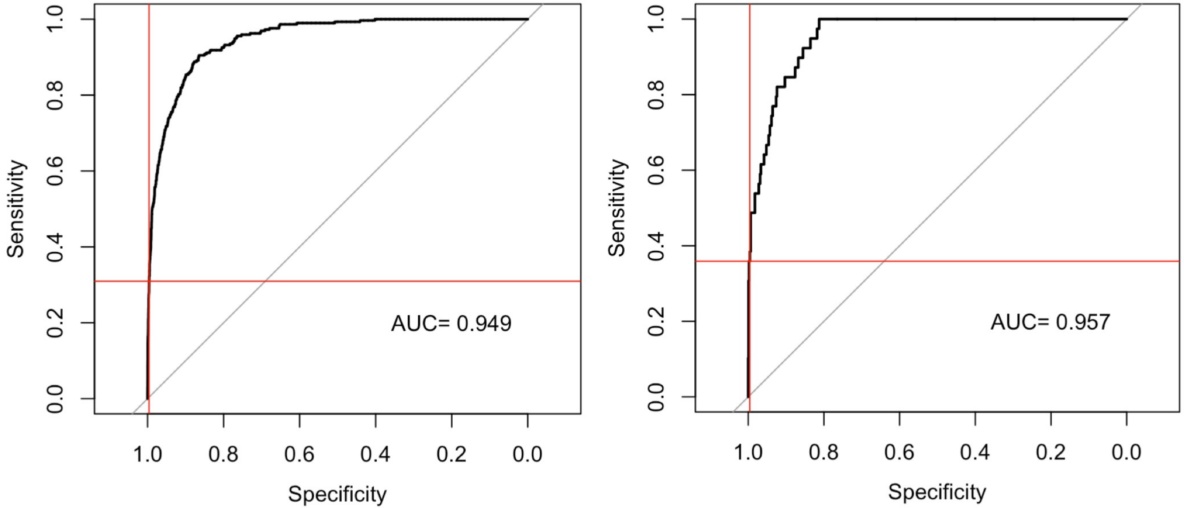
Threshold = .5

Threshold = .1

```
          Reference
Prediction  No Yes
       No  957  25
       Yes   4  14
```

```
          Reference
Prediction  No Yes
       No  900  10
       Yes  61  29
```

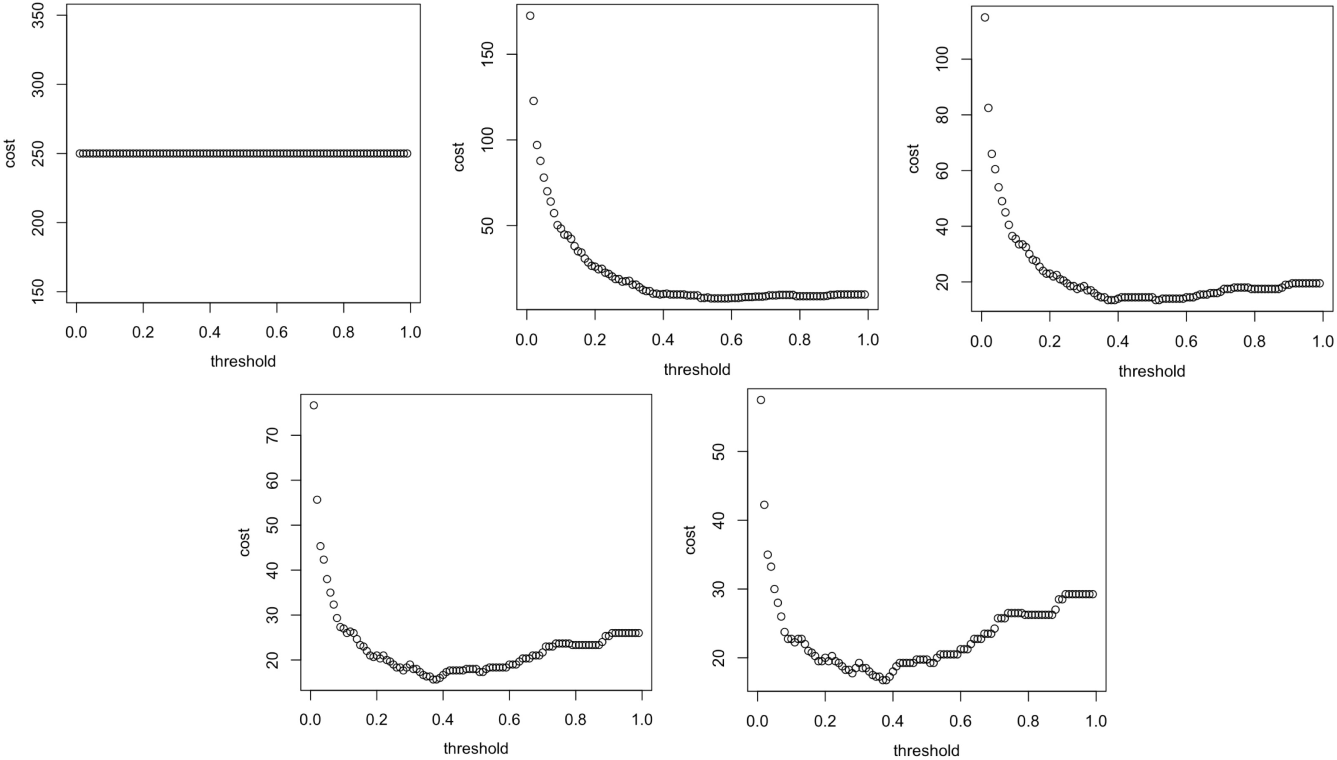
A.11 ROC


Sensitivity
1.0
0.8
0.6
0.4
0.2
0.0
Specificity
AUC= 0.949
Sensitivity
Specificity
AUC= 0.957

A.12 Real Cost Function

- What is the relative cost of
- True Positive
- True Negative
- False Positive
- False Negative

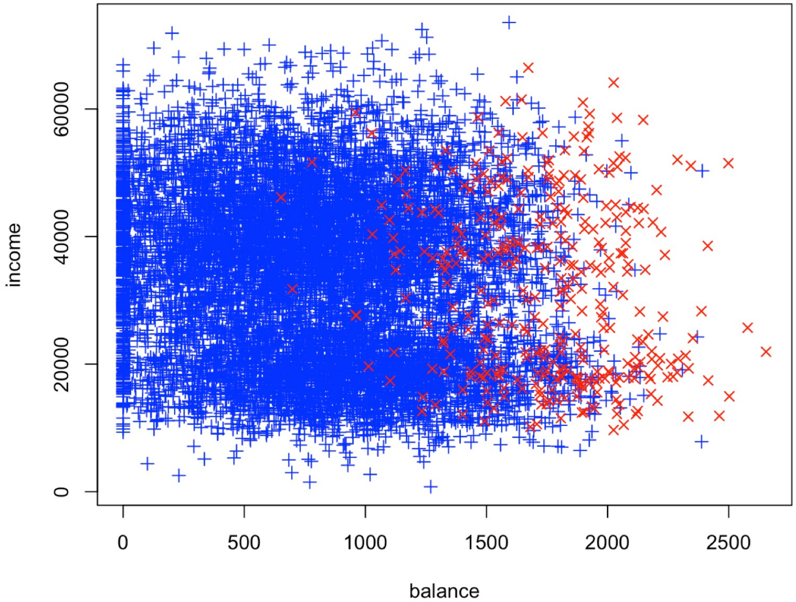
```
(TP, TN, FP, FN)
c(1,  1,  1,  1)/4
c(0,  0,  3,  1)/4
c(0,  0,  1,  1)/2
c(0,  0,  1,  2)/3
c(0,  0,  1,  3)/4
```



cost
threshold
0.0
0.2
0.4
0.6
0.8
1.0
150
200
250
300
350
50
100
20
40
60
80
30
70

A.13 K-NN

- Can we fit the same `default` data with more flexible model like K-NN using 5-fold cross validation, and AUC as a cost function?



income
balance
0
20000
40000
60000
0
500
1000
1500
2000
2500

A.14 Measure of Fit for Classification

- Since Y is all 0 or 1, its not good to use

$$\text{MSE} = E(Y - \hat{f}(X))^2$$

- Prediction Error Rate

$$\text{PER} = E\big(I(Y \neq \hat{f}(X))\big)$$

- Estimated by

$$\text{ER}_{test} = \frac{1}{n}\sum_{i=1}^{n} I(Y \neq \hat{f}(X))$$

  But to calculate ER, you must set the threshold first.

- Use AUC