Math 407

Cross-Validation for a Classification Problem

UCI Machine Learning Repository

This research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable. This study reviewed the literature and used the following 23 variables as explanatory variables:

X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.
X2: Gender (1 = male; 2 = female).
X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).
X4: Marital status (1 = married; 2 = single; 3 = others).
X5: Age (year).

X6 - X11: History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows: X6 = the repayment status in September, 2005; X7 = the repayment status in August, 2005; . . .;X11 = the repayment status in April, 2005. The measurement scale for the repayment status is: -1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.

X12-X17: Amount of bill statement (NT dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = amount of bill statement in April, 2005.

X18-X23: Amount of previous payment (NT dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005; . . .;X23 = amount paid in April, 2005.

Overview of the process of choose a model:

1. Select a few methods to try that are appropriate for the type of data you have and your (client's) needs.
2. Train models using the methods from 1. on a set of data (training set)
3. Choose the model with the best performance on a different set of data (test set) *This step is called "model selection"*
4. Get an unbiased estimate of your chosen model's performance using a third set of data (validation)

This step is called "model assessment"

Step 1: Which methods are appropriate to try?

- Linear regression
- kNN Regression
- kNN Classification
- Logistic Regression
- Linear Discriminant Analysis (i.e. Bayes Rule with normal distribution and equal variance assumption)
- Quadratic Discriminant Analysis (i.e. Bayes Rule with normal distribution)

Split entire dataset into training, selection and assessment sets

```
> dd = read.csv("default.csv", header=TRUE)
> str(dd)
'data.frame':   30000 obs. of  25 variables:
 $ ID : int  1 2 3 4 5 6 7 8 9 10 ...
 $ X1 : int  20000 120000 90000 50000 50000 50000 500000 100000 140000 20000 ...
 $ X2 : int  2 2 2 2 1 1 1 2 2 1 ...
 $ X3 : int  2 2 2 2 2 1 1 2 3 3 ...
 $ X4 : int  1 2 2 1 1 2 2 2 1 2 ...
 $ X5 : int  24 26 34 37 57 37 29 23 28 35 ...
 $ X6 : int  2 -1 0 0 -1 0 0 0 0 -2 ...
 $ X7 : int  2 2 0 0 0 0 0 -1 0 -2 ...
 $ X8 : int  -1 0 0 0 -1 0 0 -1 2 -2 ...
 $ X9 : int  -1 0 0 0 0 0 0 0 0 -2 ...
 $ X10: int  -2 0 0 0 0 0 0 0 0 -1 ...
 $ X11: int  -2 2 0 0 0 0 0 -1 0 -1 ...
 $ X12: int  3913 2682 29239 46990 8617 64400 367965 11876 11285 0 ...
 $ X13: int  3102 1725 14027 48233 5670 57069 412023 380 14096 0 ...
 $ X14: int  689 2682 13559 49291 35835 57608 445007 601 12108 0 ...
 $ X15: int  0 3272 14331 28314 20940 19394 542653 221 12211 0 ...
 $ X16: int  0 3455 14948 28959 19146 19619 483003 -159 11793 13007 ...
 $ X17: int  0 3261 15549 29547 19131 20024 473944 567 3719 13912 ...
 $ X18: int  0 0 1518 2000 2000 2500 55000 380 3329 0 ...
 $ X19: int  689 1000 1500 2019 36681 1815 40000 601 0 0 ...
 $ X20: int  0 1000 1000 1200 10000 657 38000 0 432 0 ...
 $ X21: int  0 1000 1000 1100 9000 1000 20239 581 1000 13007 ...
 $ X22: int  0 0 1000 1069 689 1000 13750 1687 1000 1122 ...
 $ X23: int  0 2000 5000 1000 679 800 13770 1542 1000 0 ...
 $ Y  : int  1 1 0 0 0 0 0 0 0 0 ...
```

How large should my Assessment set be?

Suppose we want to know the population misclassification rate to within 1% with 95% confidence.

This means we want the half-width of the CI to be 0.01:

$$0.01 = 1.96\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

Worse case is that $\hat{p} = 0.5$. *Solving for n in*

$$0.01 = 1.96\sqrt{\frac{1}{4n}}$$

yields n = $\frac{1.96^2}{4(0.01^2)}$ *= 9604*

In order to estimate the misclassification rate to within 1%, I need to have at least 9604 clients in my assessment set.

Randomly choose 9604 observations to set aside for assessing my final model…

```
> # assessment set
>
> set.seed(42)
> assessID = sample(1:nrow(dd), replace=FALSE, size=9604)
>
> ddA = dd[assessID, ]
> dim(ddA)
[1] 9604   25
>
> dd = dd[-assessID, ]
> dim(dd)
[1] 20396    25
```

I could take another 9604 out of the 20396 to use as the selection set, but I’m a bit worried about having enough training observations so I’ll use cross-validation to train and selection.

5 fold Cross Validation for training and selection

Randomly split dataset into 5 equally sized parts

*Train models on 4 of the 5 parts, get accuracy on the remaining part (i.e. test)

Repeat * five times with each part getting the chance to be the test set

Average over the 5 accuracies per model to get the CV accuracy

Choose the model with the highest CV accuracy.

Logistic Regression – payments and amount due for the last 7 months

```
> set.seed(43)
>
> foldID = sample(1:5, replace=TRUE, size = 20396)
>
>
> accLR_7months = numeric(5)
> for (k in 1:5) {
+
+  fit = glm(Y~X12+X13+X14+X15+X16+X17+X18+X19+X20+X21+X22+X23, data=dd[foldID!=k, ], family="binomial")
+
+  predProb = predict(fit, newdata=dd[foldID==k, ], type="response")
+  hatY = 1*(predProb>0.5)
+  accLR_7months[k] = mean(hatY==dd$Y[foldID==k])
+
+ }
```

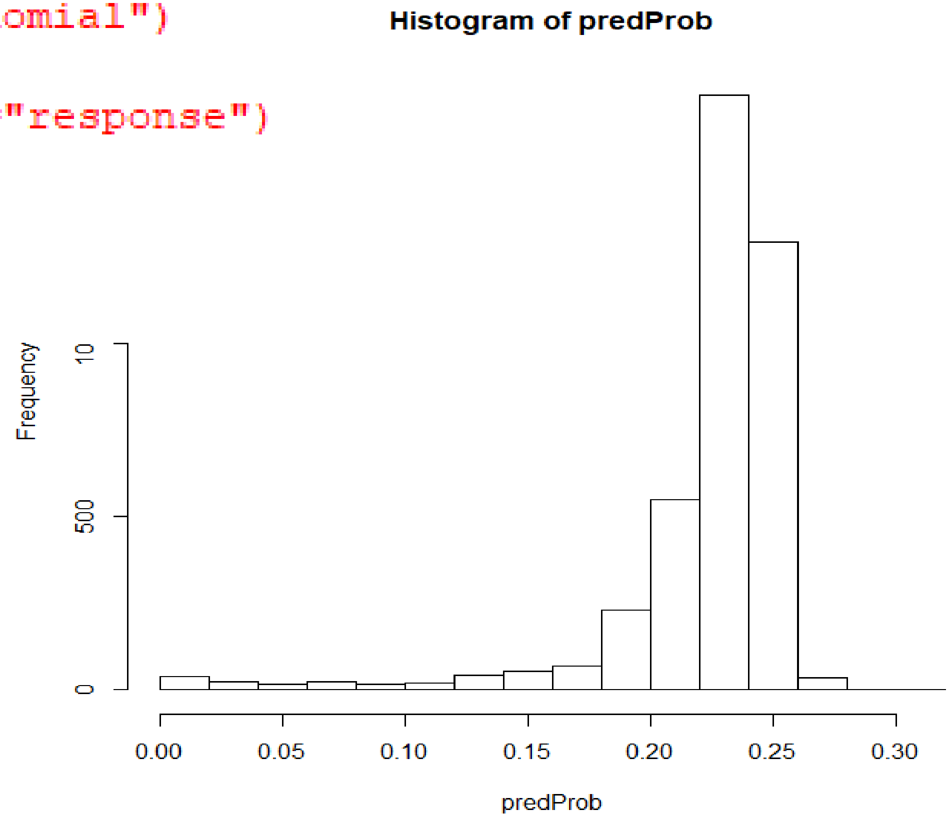
```
> accLR_7months
[1] 0.7773171 0.7768456 0.7751449 0.7802034 0.7842947
>
> mean(accLR_7months)
[1] 0.7787612
> table(dd$Y)/nrow(dd)

        0         1 
0.7789272 0.2210728 
```

Logistic Regression – payments and amount due for the last 1 month

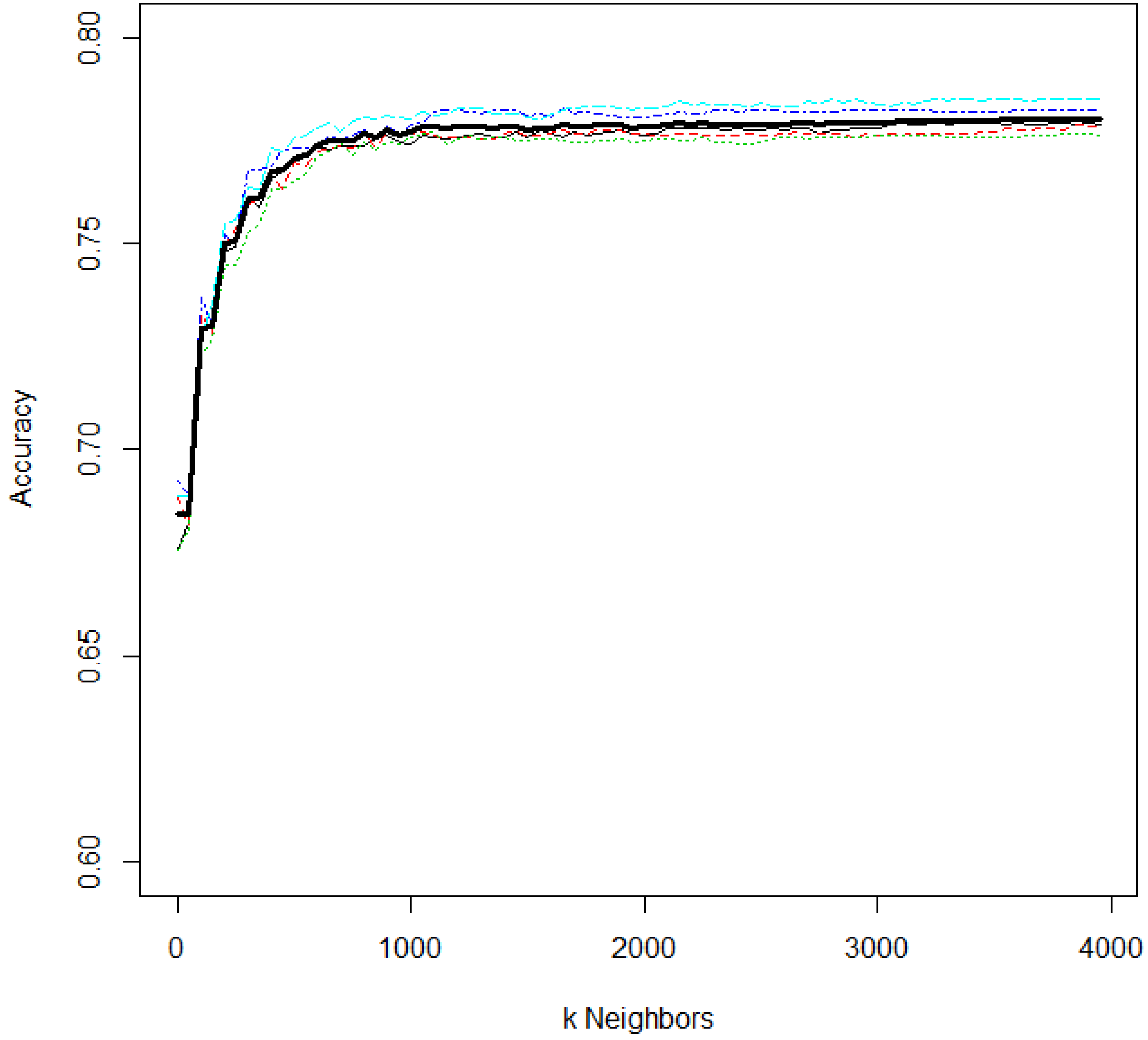
```
> accLR_1months = numeric(5)
> for (k in 1:5) {
+
+  fit = glm(Y~X12+X18, data=dd[foldID!=k, ], family="binomial")
+
+  predProb = predict(fit, newdata=dd[foldID==k, ], type="response")
+  hatY = 1*(predProb>0.5)
+  accLR_1months[k] = mean(hatY==dd$Y[foldID==k])
+
+ }

> accLR_1months
[1] 0.7780488 0.7768456 0.7751449 0.7802034 0.7842947
>
> mean(accLR_1months)
[1] 0.7789075
```

Logistic Regression – payments and amount due for the last 1 month – change cutoff

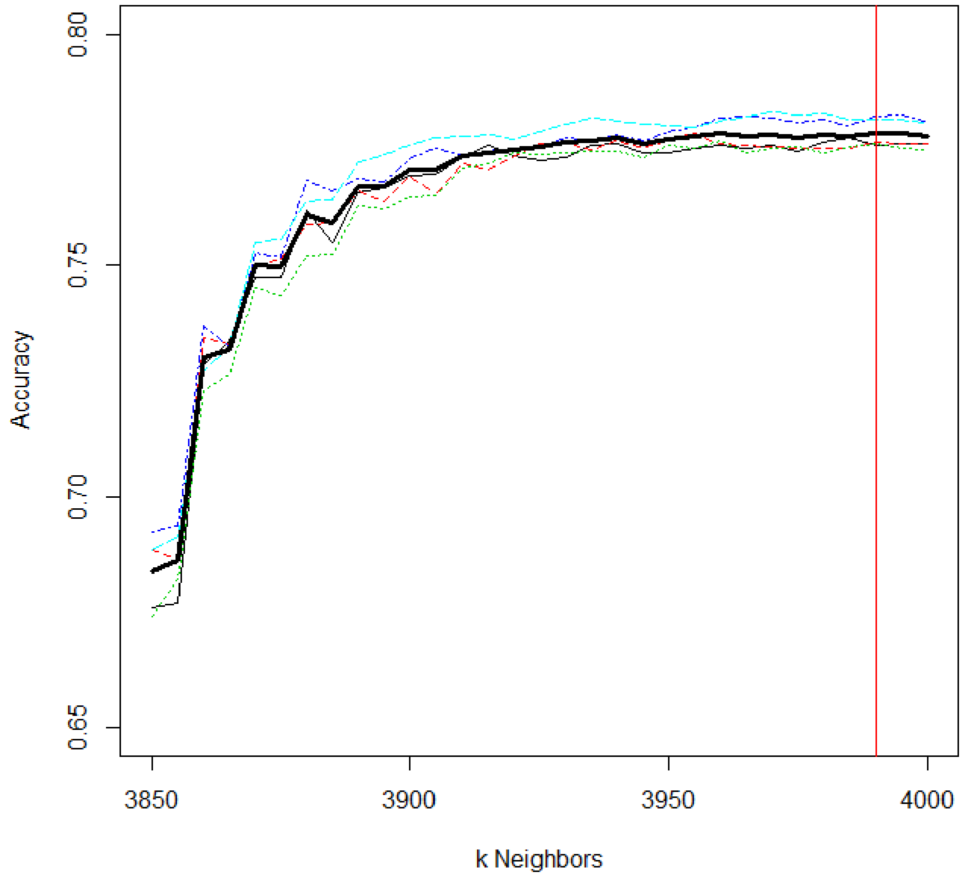
```
>
> accLR_1months = numeric(5)
> for (k in 1:5) {
+
+  fit = glm(Y~X12+X18, data=dd[foldID!=k, ], family="binomial")
+
+  predProb = predict(fit, newdata=dd[foldID==k, ], type="response")
+  hatY = 1*(predProb>0.28)
+  accLR_1months[k] = mean(hatY==dd$Y[foldID==k])
+
+ }

> accLR_1months
[1] 0.7782927 0.7768456 0.7751449 0.7804515 0.7840524
>
> mean(accLR_1months)
[1] 0.7789574
> table(dd$Y)/nrow(dd)

        0         1
0.7789272 0.2210728
```



kNN Classification – black line is average over 5 folds



Best Accuracy is 0.7784 at k=3990



Model selection and final training

Selection on overall Accuracy

Model	Overall Cross Validated Accuracy
Logistic with 7 months cutoff = 0.5	0.7788
Logistic with 1 month cutoff = 0.5	0.7780
Logistic with 1 month cutoff = 0.28	0.7789
Best kNN k=3990	0.7784

Refit the best model on all 5 folds, i.e. all data used for training and selection. This is the final model.

Model Assessment

```
> fit = glm(Y~X12+X18, data=dd, family="binomial")
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
> summary(fit)

Call:
glm(formula = Y ~ X12 + X18, family = "binomial", data = dd)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8245  -0.7372  -0.7129  -0.4676   4.9743

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.116e+00  2.197e-02  -50.81   <2e-16 ***
X12          4.052e-07  2.597e-07    1.56    0.119
X18         -3.818e-05  3.373e-06  -11.32   <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: 21549  on 20395  degrees of freedom
Residual deviance: 21305  on 20393  degrees of freedom
AIC: 21311

Number of Fisher Scoring iterations: 6

>
> predProb = predict(fit, newdata=ddA, type="response")
> hatY = 1*(predProb>0.28)
>  mean(hatY==ddA$Y)
[1] 0.7788421
```

I am 95% confident that the true accuracy of this logistic regression model is between 77.06% and 78.72%.