Jian Sun, Michael Chiang
December 15th, 2016

Analyzing Default Payments of Credit Card Clients in Taiwan

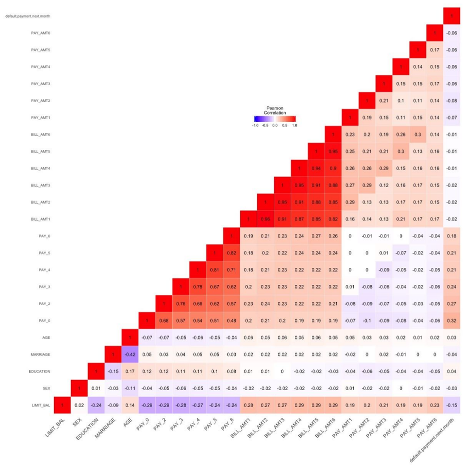
Introduction

Back in 2005, credit card issuers in Taiwan faced a cash and credit card debt crisis, with delinquency expected to peak in the third quarter of 2006 (Chou). In order to increase market share, card-issuing banks in Taiwan over-issued cash and credit cards to unqualified applicants. At the same time, most cardholders, irrespective of their repayment ability, overused credit cards for consumption purposes and accumulated heavy cash and credit card debts. This crisis caused a blow to consumer financial confidence and presented a big challenge for both banks and cardholders.

Background of Data

Our dataset, titled "Default Payments of Credit Card Clients in Taiwan from 2005," contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005. From the original 30,000 observations, we have randomly selected 5,000 for training and 5,000 for testing. Thus, we have two datasets: one to train our models, and one to test their predictive ability.

The dataset contains a total of 24 variables. Our response variable is default.payment.next.month, a categorical predictor indicating whether an individual will default on next month's payment, (1=yes, 0=no). There are four demographic predictors: SEX (1=Male, 2=Female), EDUCATION (1=Graduate School,...,6=Unknown), MARRIAGE (1-Marriage, 2=Single, 3=Others), AGE. There are three groups of financial predictors: (PAY_0, PAY_1,..., PAY_6), which indicates repayment status, (BILL_AMT1…BILL_AMT6), which indicates the amount of bill statement and (PAY_AMT1…PAY_AMT6), which indicates the amount of previous payment. Each of these groups comprise of six predictors, one for each month from April to September of 2005, in reverse order. Our last variable is LIMIT_BAL, which represents the amount of credit given in New Taiwan (NT) dollars. We noticed that PAY_0's actual range of values is from (-2, 0…8). However, the assignment claims that the range of values is from (-1, 1…9). We mitigate this problem by defining our range of values for PAY_0 as that of the assignment's values minus one. In other words, (PAY_0=-2) in our dataset represents "pay duly," (PAY_0=0) represents "payment delay for one month," and so on.



To gain an initial understanding of the relationships between our covariates, we use the “ggplot2” package in R to create a correlation heat map (the bigger one is attached in Appendix). From the plot above, we see that the correlation between our response variable and the remaining 23 covariates is not very strong. As a result, we proceed to explore this relationship by utilizing the classification techniques we have studied this semester.

Data Analysis

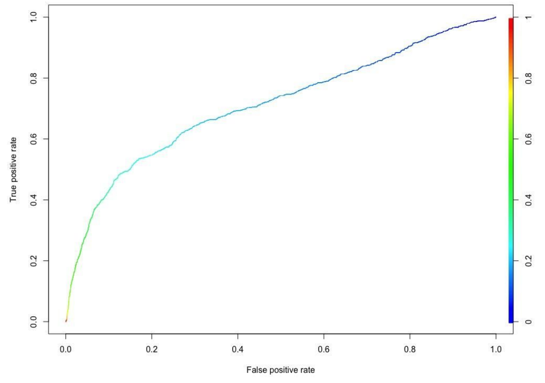
We now move on to our analysis of the dataset, where we used R programming to conduct a total of eight classification techniques.

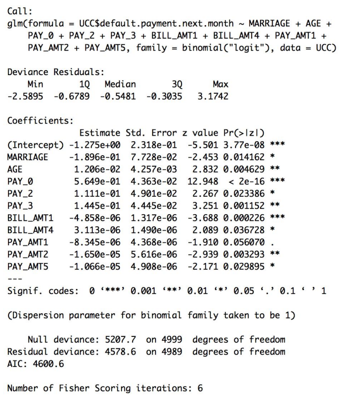
1. Logistic Regression

We begin with logistic regression, which we selected because it allows us to explore the association between our dichotomous response variable and the covariates. Logistic regression has three major assumptions: the response variable should be binary, there should be no outliers in the dataset, and there should be no high intercorrelations among the predictors. In both the training and testing datasets, we treat the response variable (default.payment.next.month), and the unordered demographic variables (sex, marriage) as factors.

To conduct our analysis, we used the "glm" and "drop1" functions found in the "ElemStatLearn" and "ROCR" packages in R. The “drop1” function allows us to efficiently select the best predictors for our logistic model. One by one, the function drops the variable with the highest p-value. We are left with 10 variables: MARRIAGE, AGE, PAY_0, PAY_2, PAY_3, BILL_AMT1, BILL_AMT4, PAY_AMT1, PAY_AMT2 and PAY_AMT5.

With these 10 variables, we proceed to build our logistic model using our training dataset. We then test our model using the testing dataset to see how accurately our model predicts fits the response values. Using the “prediction” and “performance” functions in R, we obtain a relative mean squared error (MSE) of 0.146 and plot our receiver operating characteristic (ROC) curve.





```
Call:
glm(formula = UCC$default.payment.next.month ~ MARRIAGE + AGE +
    PAY_0 + PAY_2 + PAY_3 + BILL_AMT1 + BILL_AMT4 + PAY_AMT1 +
    PAY_AMT2 + PAY_AMT5, family = binomial("logit"), data = UCC)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5895  -0.6789  -0.5481  -0.3035   3.1742

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.275e+00  2.318e-01  -5.501 3.77e-08 ***
MARRIAGE    -1.896e-01  7.728e-02  -2.453 0.014162 *
AGE          1.206e-02  4.257e-03   2.832 0.004629 **
PAY_0        5.649e-01  4.363e-02  12.948  < 2e-16 ***
PAY_2        1.111e-01  4.901e-02   2.267 0.023386 *
PAY_3        1.445e-01  4.445e-02   3.251 0.001152 **
BILL_AMT1   -4.858e-06  1.317e-06  -3.688 0.000226 ***
BILL_AMT4    3.113e-06  1.490e-06   2.089 0.036728 *
PAY_AMT1    -8.345e-06  4.368e-06  -1.910 0.056070 .
PAY_AMT2    -1.650e-05  5.616e-06  -2.939 0.003293 **
PAY_AMT5    -1.066e-05  4.908e-06  -2.171 0.029895 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5207.7  on 4999  degrees of freedom
Residual deviance: 4578.6  on 4989  degrees of freedom
AIC: 4600.6

Number of Fisher Scoring iterations: 6
```

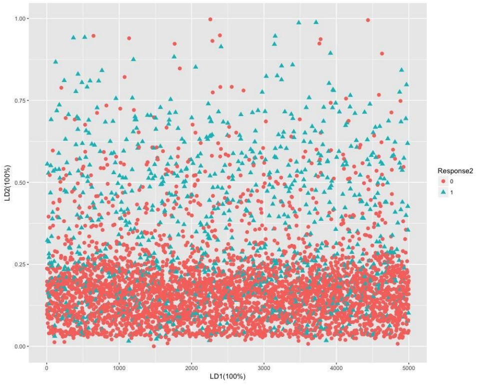
With our ROC plot, we can simply calculate the area under the curve (AUC) to obtain the correct classification rate of our model. In this case, the AUC is 0.7123, which means our model correctly classifies an individual’s default payment status 71.23% of the time. Note that we can simply calculate the misclassification rate by subtracting the classification rate from 100, which is 19.77% in this model.

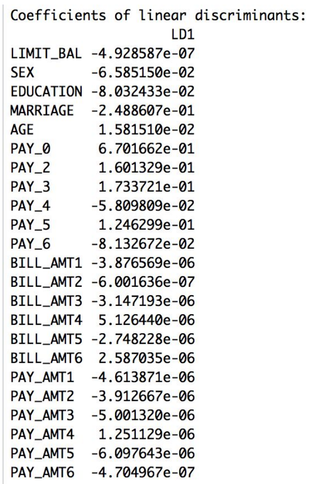
2. Linear Discriminant Analysis (LDA)

In our second method, we conduct discriminant analysis using the “lda” function from the “MASS” package in R. LDA allows us to determine which of the covariates are the best predictors of an individual's probability of default payment. LDA assumes that the conditional probability density functions of our response variable, p(x|y=0) and p(x|y=1), are normally distributed and that the covariances of each category are identical and fully ranked.

To begin, we treat our response variable, default.payment.next.month, as a factor in both the training and testing datasets. The remaining 23 covariates will serve as our predictors. Similar to our first method, we use the training dataset to build our model, then assess its predictive ability using the testing dataset. Utilizing the “predict” and “sum” functions, we obtain a relative MSE of 1.0923. Next, we use the “table” function to build a classification table and obtain the correct classification rate by adding up the terms on the diagonal. From the table, we obtain a classification rate of 0.809. In other words, our discriminant analysis model accurately classifies the default payment status of an individual 80.9% of the time.

The red point and green points in the plot above represent the two classes, 0 or 1, of the response variable, default.payment.next.month. Red indicates (0=no), which means that the individual will make their payment on time. Green indicates (1=yes), which means that the individual will default on their payment. As you can see, most of the points are red.



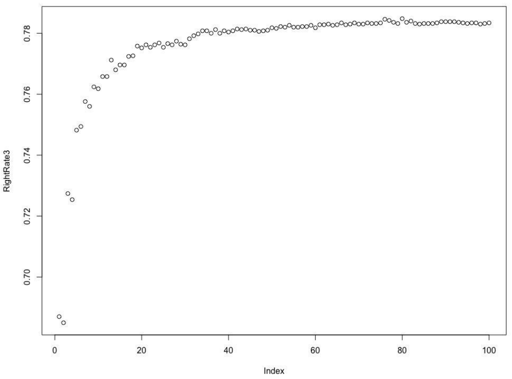


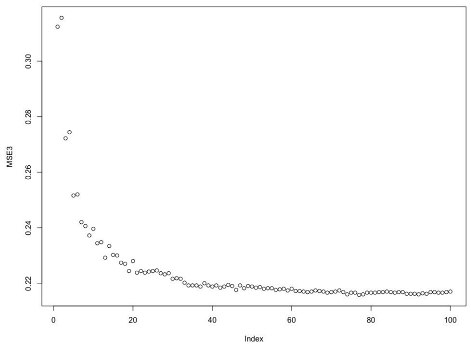
```
Coefficients of linear discriminants:
                    LD1
LIMIT_BAL -4.928587e-07
SEX       -6.585150e-02
EDUCATION -8.032433e-02
MARRIAGE  -2.488607e-01
AGE        1.581510e-02
PAY_0      6.701662e-01
PAY_2      1.601329e-01
PAY_3      1.733721e-01
PAY_4     -5.809809e-02
PAY_5      1.246299e-01
PAY_6     -8.132672e-02
BILL_AMT1 -3.876569e-06
BILL_AMT2 -6.001636e-07
BILL_AMT3 -3.147193e-06
BILL_AMT4  5.126440e-06
BILL_AMT5 -2.748228e-06
BILL_AMT6  2.587035e-06
PAY_AMT1  -4.613871e-06
PAY_AMT2  -3.912667e-06
PAY_AMT3  -5.001320e-06
PAY_AMT4   1.251129e-06
PAY_AMT5  -6.097643e-06
PAY_AMT6  -4.704967e-07
```

3. K-Nearest Neighbor (k-NN) Classification

Our next method is k-NN regression, where we use the “knn” function from the “class” package in R. We chose this method because individuals with similar financial details have similar credit ratings. Using the k-NN algorithm is an efficient way to predict a customer's credit rating based on our existing dataset. k-NN assumes that the data points are in a metric space, that each of the training datum consists of a set of vectors and class label associated with each vector, and that the value k decides how many neighbors influence the classification.

We treat the response variable (default.payment.next.month), and the unordered demographic variables (sex, marriage) as factors in both our datasets. The remaining 23 variables serve as our predictors. We use the training dataset to build our model, then assess its predictive ability using the testing dataset. One of the most important parts of k-NN classification is to find an appropriately small value for k to ensure that computational costs are not too high. To do this, we ran a loop in R, setting the range of k from 1 to 100, which gave us the plot below on the left.





We find that k = 7 corresponds to a decent classification rate. We also calculate a relative MSE of 0.242 using the “knn” and “sum” functions. The plot on the right above depicts the MSE for each k. With k = 7, we create a classification table and obtain a classification rate of 0.758, or 75.8%.

4. Support Vector Machine (SVM) Classification

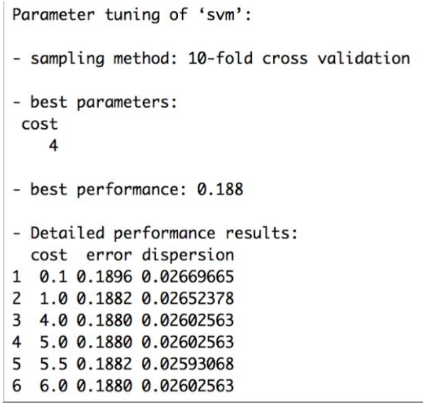
Our next method is SVM classification, which was first introduced in 1992 by Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik. Given training data, the SVM algorithm outputs an optimal hyperplane which categorizes new examples. The goal is to find a hyperplane that gives the largest minimum distance to the training examples. This distance, on both side of the hyperplane, is called the margin. This method involves some very complex transformations on the data, and figures out how to separates the dataset based on the categories of our response. We chose SVM because it allows us to capture complex relationships in the data without performing the transformations manually.

SVM seeks to solve the following optimization problem:

$$\begin{aligned}&\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M\\&\text{subject to } \sum_{j=1}^{p}\beta_j^2 = 1,\\&y_i(\beta_0+\beta_1x_{i1}+\beta_2x_{i2}+\ldots+\beta_px_{ip}) \geq M(1-\epsilon_i),\\&\epsilon_i \geq 0,\ \sum_{i=1}^{n}\epsilon_i \leq C,\end{aligned}$$

Here, M is the width of the margin, and C is a nonnegative tuning parameter. We can think of C as a budget for the amount that the margin can be violated by the n observations (James).

To conduct SVM classification in R, we utilize the “tune” function from the “e1071” package. We treat the response variable (default.payment.next.month), and the unordered demographic variables (sex, marriage) as factors in both the training and testing datasets. The remaining 23 variables serve as our predictors. To begin, we need to find an adequate value for the tuning parameter C, labeled “cost” in R. We use the “tune” function to conduct 10-fold cross validation (CV) in order to obtain the best cost among (0.001, 0.01, 0.1, 1, 5, 10, 100). Among these values, we find that cost=5 gives us the lowest error of 0.1878. In an attempt to lower this error further, we conduct 10-fold CV again, this time on (0.1, 4, 5, 5.5, 6). We find that cost=4 gives us an error of 0.188. The final cross validation results from R are shown below on the left.



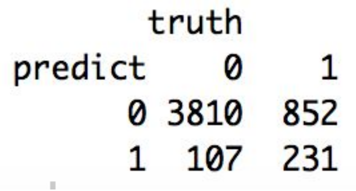
```
Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
 cost
    4

- best performance: 0.188

- Detailed performance results:
  cost  error dispersion
1  0.1 0.1896 0.02669665
2  1.0 0.1882 0.02652378
3  4.0 0.1880 0.02602563
4  5.0 0.1880 0.02602563
5  5.5 0.1882 0.02593068
6  6.0 0.1880 0.02602563
```



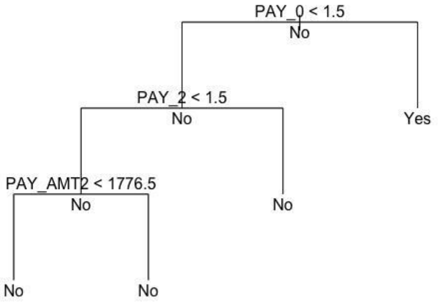
```
          truth
predict     0    1
      0  3810  852
      1   107  231
```

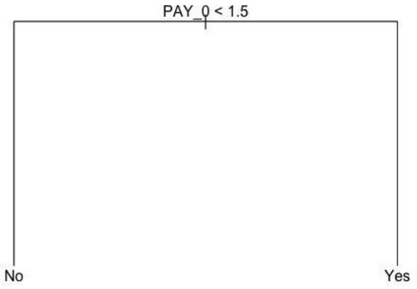
Now, we proceed to build the SVM model with 23 covariates and a cost value of 4 using the training dataset. We then test our model using the testing dataset to see how accurately our model predicts fits the response values. We obtain a relative MSE of 0.1918. We then create a classification table, shown above on the right, which gives us a correct classification rate of 0.8082, or 80.82%.

5. Classification Tree

Next, we use classification trees to build a model. Classification trees provide a visual representation of our statistical problem and illustrates all factors within our dataset that are considered relevant to the response. It also closely mirrors human decision making compared to other methods. This technique makes no assumptions about the data.

In both the training and testing datasets, we treat the response variable (default.payment.next.month), and the unordered demographic variables (sex, marriage) as factors. Utilizing the “tree” function, we use the 23 covariates from the training dataset to build our model. Prior to pruning, the initial tree had four terminal nodes, consisting of three variables: PAY_0, PAY_2 and PAY_AMT2. This figure is shown below on the left.





After pruning, we are left with only one predictor, PAY_0. This tree can be seen above on the right. We test our model using the testing dataset, and create a classification table. Our calculations give us a classification rate of 0.8154, or 81.54%.

Conclusion

Of the five classification techniques, only the logistic, linear discriminant analysis, and classification tree models allow us to analyze variable significance. Consequently, we assess the importance of our covariates using these three methods.

From the summary of our logistic model, we see that MARRIAGE and AGE are among the 10 variables used to build our logistic model. MARRIAGE has a negative coefficient of -1.896e-01, which tells us that the variable lowers an individual's chance of defaulting on his/her next payment. For example, individuals with a marriage status of (3=others) have a higher probability of making their payment on time, as opposed to individuals who are (1=married). AGE, on the other hand, has a positive coefficient of 1.206e-02, which means that the variable increases an individual's chance of defaulting. In particular, the older an individual is, the higher the probability of defaulting.

From the summary of our LDA model, we see that all four demographic variables: SEX, EDUCATION, MARRIAGE, and AGE have relatively large coefficients. SEX and EDUCATION have negative coefficients of -6.5815e-02 and -8.0324e-02, respectively. This means that both variables lower the probability of an individual defaulting. In particular, (female=2) subjects and individuals with (unknown=6) education level are more likely to make their payments on time.

From our classification tree model, both trees generated before and after pruning contain no demographic variables. This suggests that the demographic covariates were not considered relevant when predicting an an individual's probability of default payment.

From our three models, PAY_0 is universally recognized as the best predictor of default payment. Its corresponding coefficient estimate is the largest among all variables. Furthermore, the final classification tree model selected PAY_0 as the only relevant variable in predicting our response. Our two models, logistic and LDA, agreed that four other variables were important in predicting default payment: MARRIAGE, AGE, PAY_2, PAY_3.

Now that we have analyzed the variables in our dataset, we can assess the overall predictive ability of our five statistical methods. Of the five approaches, the classification tree model produced the highest correct classification rate of 0.8154. This tells us that our model correctly classifies an individual's default payment status 81.54% of the time, based on the testing dataset. This indicates that our model is a good fit for the datasets. After pruning, our final classification tree only contains one terminal node, with the PAY_0 variable. If the value of PAY_0, which represents repayment status in September 2005, is larger than 1.5, then our model predicts that the individual will default on his/her payment next month. If the value of PAY_0 is less than 1.5, then our model forecasts that the individual's payment for the next month will be made on time.

References

1. Chou, M. (2006). Cash and credit card crisis in Taiwan. Business Weekly, 24–27.

2. James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. An Introduction to Statistical Learning: With Applications in R. New York: Springer, 2013.

3. Lecture notes, slides and handouts from STAT 6242, taught by Efstathia Bura

4. Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

5. Yeh, I-Cheng, and Che-Hui Lien. "The Comparisons of Data Mining Techniques for the Predictive Accuracy of Probability of Default of Credit Card Clients." Expert Systems with Applications 36.2 (2009): 2473-480.