Math437 – Credit Card Defaults

Hunter Thompson

May 4th, 2020

Introduction

What is "Credit Defaulting"?

Defaulting is the result of failing to pay a loan. To the average person, this would consist of not paying off everyday items, such as a credit card, or any student loans. Generally, as the customer for a loan, defaulting will negatively affect credit score, and will likely lead to rejection for any future loans one may try to leverage.

From the provider standpoint, a default means the loan provider is not getting paid. Given that providing loans is an investment, with an initial amount supplied, and interest collected over a given time period, the supplier wants to secure this investment from the risk of defaulting. An established customer may have a high credit score, and thus a lower risk for defaulting compared to some, but the concern is how does a loan provider minimize this risk, particularly when there is minimal history in a credit score?

The Dataset

The *Default of Credit Card Clients* dataset, provided by University of California at Irvine's Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients) takes 30,000 Taiwanese adults, ranging in age from 21 to 79 years old, and attempts to classify them based off of 23 total variables. These variables seem to best match four categories.

The target data for the set is a 0 or 1 classification if they will default. A value of 1 means the client has a significant probability of defaulting on the next payment, while a 0 represents a safer client. There are a total of 23,364 clients unlikely to default, and a further 6,636 likely to default on the next payment.

The first set of variables provides some basic information about the individual. This ranges from the amount of initial credit provided, to their gender, or even their education level. It also assigns a unique ID to each individual, however this ID was excluded from any models.

The remaining sets cover a set time, from April 2005 through September 2005, and all monetary values are in the New Taiwan Dollar.

The second set of variables is the timeliness of past payments through the given months. The tardiness is rounded to the month, ranging from 1 to 9+ months. An on-time payment is provided with a value of -1.

The third set of variables represents the total bill statement for the month. Given that it covers the same time range, it demonstrates the amount of money charged to the card for the time period it represents.

The fourth variable in the dataset is the amount paid by the client that month. This, in conjunction with the total bill statement for the month gives a relatively strong basis for the frequency of payments in comparison to the overall usage of a card.

Full-Variable Models

The Full-Variable models were applied with all the variables in the dataset to provide an overarching view of how the variables relate to the target. Given that it is a classification dataset, the two models used were a Logistic Regression Model, and a K-Nearest Neighbors Model. These both were scored appropriately.

Logistic Regression

The first trial of the LR model seemed to perform reasonably well. Coefficients in the model were small, which appeared reasonable given that it was attempting to predict values from 0 to 1. To confirm the initial appearance, the model underwent a Leave-One-Out Cross-Validation.

LOOCV for Full-Variable Logistic Regression shows the performance of the model is decent, but not quite the accuracy needed for a large scale operation. Several trials showed that the error percentage ranged from about 0.2%, up to around 3%. Additionally, several trials would demonstrate the predicted value would be 2, in part due to the poor dataset. These predictions were subsequently removed from trial attempts.

K-Nearest Neighbors

For the KNN models, the credit data was split 80-20 into training and testing data, randomly separated, but with the same set seed for reproducibility. With 24,000 in the training data, the initial model was trained with k=3. Accuracy was determined by dividing the correct classifications by the total number of samples. For a model with k of 3, the accuracy was approximately 73.37%.

This model and accuracy scoring was then repeated for k-values in the range of 1 through 10. Across all 11 models, the mean accuracy was 74.17%, with a standard deviation of 2.56. The highest accuracy occurred with a k-value of 9, predicting 76.63% correct.

Best Subset Models

To select the best subset, all variable were tested using the Best Subset algorithm, and then the same Logistic Regression and K-Nearest Neighbor Models were simulated.

Best Subset Selection

Running the data through the best subset selection process gave varied results. Across the 23 total models created, the four chosen scoring methods disagreed. Given that R^2 will commonly select the model with the most variables, it is unsurprising that the 23-variable model scored the best, achieving a value of 0.12401. BIC selected the 9-variable model as the best, providing a score of -3832.02. Adjusted R^2 and C_P selected the same model as the ideal one. Given the 15-variable model, it scored as 0.12349 and 11.085 respectively. With both Adjusted R^2 and C_P selecting the same one, the following models were run with the 15-variable dataset. The best subset variables are available to see in the appendix.

Logistic Regression

The second model of Logistic Regression shared similar results to that of the Full-Variable one, in that the coefficients for the regression are all small. This too corresponds to the understanding that the prediction value is 0 or 1, and represents a probability.

Similar to the LOOCV previously run, performances were okay, but not successful enough to ensure that an LR model would accurately represent the probability for defaulting on credit loans. The several trials run for misclassification on the LR model averaged to about 0.4% calculating incorrectly, but occasionally reaching up to 2% in errors. Additionally, there was the occasional value '2' represented in the target data, in part due to the poor dataset.

K-Nearest Neighbors

The KNN model for the best subset data performed slightly worse overall, in comparison to the full-variable model did. Running the same process for k-values from 1 through 10, the mean accuracy was 73.84%, with a standard deviation of 2.73. The highest performer, however, did show an improvement to 76.77%, a 0.002% increase in performance. This difference is negligible.

Conclusion

Results

Ultimately, neither Logistic Regression nor K-Nearest Neighbors performed at a level to demonstrate effective prediction models.

Logistic Regression showed the weaknesses in the dataset. With poor records, it is difficult to establish a consistent model in terms of predicting default probability. This is most apparent for ID 24, when compared to ID 34. Both have a similar history when it comes to payments, valued of -2 repeatedly, discussed in the limitations in the model section later, however the target data is shown to be different by the dataset. Given the difficulty in cleaning the data before usage, it is unrealistic to expect perfect results from the model. However, despite that issue, the

KNN seemed like it would provide semi-reliable results, but with a best accuracy of around 77%, for a large money operation like credit loans, it would be unwise to risk any significant amount of money with a 23% chance of incorrect results. Additionally, the models favored erring on the side of a client being unlikely to default on their credit loans. This means that using KNN as a model for determining the risk of a client is faulty, and should not be relied upon to make decisions.

Best Subset, in particular the 15-variable model, favored recent months when it came to deciding the best variables to include in a model. The decision seems reasonable, as it is more probable that someone already not paying off the bill, or someone making larger transactions in recent months, is more at risk for defaulting on their payments. A curious side note is that for the timeliness of payment variables in the model selected, best subset recommended that the month of May to be included, but not April nor June. Additionally, it seems that the total bill had minimal impact on the likeliness of defaulting, when respect to the best subset selection process.

Limitations in the Model

Overall, the biggest limitation to the model is the target data used a prediction algorithm to determine the likelihood of defaulting. The dataset used an Artificial Neural Network to generate the target data, and then evaluated its results using a linear regression to compare the models prediction capabilities to the real probability of defaulting for a given individual. The results of the test do seem to be promising, with a regression coefficient close to 1, and the dataset concludes that the ANN is the only accurate predictor of defaulting. To achieve these results, the dataset chose to limit the variables to those listed, as they performed the best for the ANN model used. Ideally, for the models used in this research, having access to more variables and limiting those to maximize this model would be best.

The other limitation of the models occurs during the best subset selection. Given that this data is a classification set, the scoring for any subset selection did not perform well. The scoring for all subsets never surpassed an R^2 value above 0.1241, while the values for BIC and C_P were approximately -3,832 and 11.085 respectively. Given that these scores are quite low, it demonstrated that the Best Subset Selection process did not benefit the models like they should, although the results were similar.

The final limitation of the data is the collection. Having touched upon the method used to create the target data, an artificial neural network, the collection of variables themselves are not perfect. Having explained how the variables are set up, with full explanations in the appendix, there are some deviations from the established standard. For example, in the history of past payments, there are some data points valued at -2, not a value explained in the dataset. For some instances, it represents a missing value, which is also sometimes presented as 0, but for other cases, it establishes that this person paid their bill earlier than required. In the same manner, it is very difficult to determine if the payment amount of an individual for a month is due to a lack of payment, or if their credit loan has not been created as of yet. With that concern in mind, it is difficult to differentiate between missing data values, and those of real data points. To simplify things, no clients were removed from the data, however the data should be analyzed with that in mind. This most manifests in the occasional ‘2’ that appears in the logistic regression model predictions, and should be considered when attempting to verify the accuracy of the models.

Conclusion

With the poor performances of the models, and with the inconsistent recording of the data within the UCI Repository, it is difficult to recommend these models for classifying risk of individuals defaulting on credit loans. With that in mind, the trend across all the models is that risk does seem related to payment history long term, as is consistent with credit score. With more consistent data, and better models, it may be possible to determine the risk of defaulting.

Appendix

Links

GitHub Repository: https://github.com/Hunter-Thompson037/MATH437-credit-card-defaults

UCI ML Repository: http://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

Defaulting: https://www.investopedia.com/terms/d/default2.asp

Introduction

Variable	Units (Type)	Description
Y	Class	0 or 1; likelihood of defaulting on next payment as predicted by ANN
X1	NT $	The credit amount given, for individual and family
X2	Gender	1 or 2; Male/Female
X3	Education	1-4; Graduate School/University/High School/Other
X4	Marital	1-3; Married/Single/Other
X5	Years	Age of individual

Variable	Units (Type)	Description
X6	Months	-1,1-9; payment status for the month of September 2005
X7	Months	-1,1-9; payment status for the month of August 2005
X8	Months	-1,1-9; payment status for the month of July 2005
X9	Months	-1,1-9; payment status for the month of June 2005
X10	Months	-1,1-9; payment status for the month of May 2005
X11	Months	-1,1-9; payment status for the month of April 2005

Variable	Units (Type)	Description
X12	NT $	Total bill statement for the month of September 2005
X13	NT $	Total bill statement for the month of August 2005
X14	NT $	Total bill statement for the month of July 2005
X15	NT $	Total bill statement for the month of June 2005
X16	NT $	Total bill statement for the month of May 2005
X17	NT $	Total bill statement for the month of April 2005

Variable	Units (Type)	Description
X18	NT $	Amount paid in the month of September 2005
X19	NT $	Amount paid in the month of August 2005
X20	NT $	Amount paid in the month of July 2005
X21	NT $	Amount paid in the month of June 2005
X22	NT $	Amount paid in the month of May 2005
X23	NT $	Amount paid in the month of April 2005

Full-Variable Models

Logistic Regression Coefficients

(Intercept)	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13
-0.7327393	-7.513371e-07	-0.1222122	-0.09104343	-0.1427885	0.007618866	0.5951185	0.07056818	0.09440915	0.01816162	0.03284148	0.007148338	-4.3352e-06	1.558347e-06

X14	X15	X16	X17	X18	X19	X20	X21	X22	X23
1.402869e-06	-1.150186e-07	-5.051491e-07	1.050989e-06	-1.536793e-05	-7.462898e-06	-1.833763e-06	-2.200401e-06	-4.357155e-06	-2.139983e-06

KNN 3-K Accuracy Table

kmodel_3	0	1
0	4099	1035
1	563	303

KNN 1-10 Accuracy Percentages

1.00000	2.00000	3.00000	4.0	5.0	6.00	7.00000	8.00000	9.00000	10.0
69.78333	69.38333	73.38333	73.2	75.3	75.25	76.31667	76.06667	76.63333	76.4

Best Subset Models

Subset Scoring – R^2, Adj R^2, BIC, C_P

```
      [1,] 0.1054910 0.1054611 -3323.790 613.91002
      [2,] 0.1121900 0.1121308 -3538.999 386.66999
      [3,] 0.1176986 0.1176104 -3715.411 200.16710
      [4,] 0.1189420 0.1188245 -3747.409 159.61978
      [5,] 0.1199585 0.1198118 -3771.734 126.83298
      [6,] 0.1211089 0.1209331 -3800.666  89.46753
      [7,] 0.1216015 0.1213965 -3807.175  74.61190
      [8,] 0.1222480 0.1220138 -3818.953  54.48945
      [9,] 0.1229316 0.1226683 -3832.018  33.09671
     [10,] 0.1232089 0.1229165 -3831.195  25.60768
     [11,] 0.1234600 0.1231385 -3829.480  19.01400
     [12,] 0.1236450 0.1232943 -3825.506  14.68144
     [13,] 0.1237607 0.1233808 -3819.155  12.72470
     [14,] 0.1238654 0.1234563 -3812.432  11.14053
     [15,] 0.1239255 0.1234872 -3804.180  11.08522
     [16,] 0.1239523 0.1234848 -3794.790  12.16712
     [17,] 0.1239766 0.1234799 -3785.313  13.33577
     [18,] 0.1239970 0.1234711 -3775.704  14.63655
     [19,] 0.1240104 0.1234553 -3765.854  16.17795
     [20,] 0.1240130 0.1234286 -3755.634  18.08850
     [21,] 0.1240145 0.1234008 -3745.375  20.03909
     [22,] 0.1240153 0.1233724 -3735.093  22.01233
     [23,] 0.1240156 0.1233435 -3724.796  24.00000
```

Best Subset Variable – 15 Variable Model

X1, X2, X3, X4, X5, X6, X7, X8, X10, X12, X13, X18, X19, X21, X22

Logistic Regression Coefficients

```
  (Intercept)          X1          X2          X3          X4          X5          X6          X7          X8         X10         X12         X13
-6.936483e-01 -8.050084e-07 -1.055632e-01 -1.032545e-01 -1.556523e-01  7.537375e-03  5.815686e-01  8.111323e-02  8.496816e-02  5.689725e-02 -5.804752e-06  4.395381e-06
         X18         X19         X21         X22
-1.502142e-05 -8.295116e-06 -3.601757e-06 -3.540721e-06
```

KNN 1-10 Accuracy Percentages

```
  1.0   2.0   3.00  4.00000  5.00  6.00000  7.00000  8.00000  9.00000 10.00000
68.9 68.8 73.35 73.18333 74.55 74.83333 75.96667 75.63333 76.76667 76.38333
```

LR Code Sample

```
#Logistic Regression Model
log_reg <- multinom(credit_train$Y~.,data=credit_train,trace=FALSE)
lr_coef <- summary(log_reg)$coefficients
```

KNN Code Sample

```
#KNN Model

#run the model, to start k=3
kval=3

kmodel_3 <- knn(credit_train,credit_test,cl=credit_data[random,24],k=kval)

results_3 <- table(kmodel_3,credit_data[-random,24])
```

Accuracy Function Code

```
#function to calculate accuracy of the KNN model
accuracy <- function(x) {sum(diag(x)/sum(rowSums(x))) * 100}
```