Predicting Loan Repayment

Homework due Oct 13, 2020 07:59 +08 Past due

Predicting loan repayment

In the lending industry, investors provide loans to borrowers in exchange for the promise of repayment with interest. If the borrower repays the loan, then the lender profits from the interest. However, if the borrower is unable to repay the loan, then the lender loses money. Therefore, lenders face the problem of predicting the risk of a borrower being unable to repay a loan.

To address this problem, we will use publicly available data from LendingClub.com, a website that connects borrowers and investors over the Internet. This dataset represents 9,578 3-year loans that were funded through the LendingClub.com platform between May 2007 and February 2010. The binary dependent variable **not.fully.paid** indicates that the loan was not paid back in full (the borrower either defaulted or the loan was "charged off," meaning the borrower was deemed unlikely to ever pay it back).

To predict this dependent variable, we will use the following independent variables available to the investor when deciding whether to fund a loan:

- **credit.policy**: 1 if the customer meets the credit underwriting criteria of LendingClub.com, and 0 otherwise.
- **purpose**: The purpose of the loan (takes values "credit_card", "debt_consolidation", "educational", "major_purchase", "small_business", and "all_other").
- **int.rate**: The interest rate of the loan, as a proportion (a rate of 11% would be stored as 0.11). Borrowers judged by LendingClub.com to be more risky are assigned higher interest rates.
- **installment**: The monthly installments ($) owed by the borrower if the loan is funded.
- **log.annual.inc**: The natural log of the self-reported annual income of the borrower.
- **dti**: The debt-to-income ratio of the borrower (amount of debt divided by annual income).
- **fico**: The FICO credit score of the borrower.
- **days.with.cr.line**: The number of days the borrower has had a credit line.
- **revol.bal**: The borrower's revolving balance (amount unpaid at the end of the credit card billing cycle).
- **revol.util**: The borrower's revolving line utilization rate (the amount of the credit line used relative to total credit available).
- **inq.last.6mths**: The borrower's number of inquiries by creditors in the last 6 months.
- **delinq.2yrs**: The number of times the borrower had been 30+ days past due on a payment in the past 2 years.
- **pub.rec**: The borrower's number of derogatory public records (bankruptcy filings, tax liens, or judgments).

Problem 1.1 - Preparing the Dataset

1 point possible (graded)

Load the dataset loans.csv into a data frame called loans, and explore it using the str() and summary() functions.

What proportion of the loans in the dataset were not paid in full? Please input a number between 0 and 1.

Answer: 0.1600543

Explanation

From table(loans$not.fully.paid), we see that 1533 loans were not paid, and 8045 were fully paid. Therefore, the proportion of loans not paid is 1533/(1533+8045)=0.1601.

Submit You have used 0 of 3 attempts

Calculator

Answers are displayed within the problem

Problem 1.2 - Preparing the Dataset

1 point possible (graded)

Which of the following variables has at least one missing observation? Select all that apply.

- [ ] credit.policy
- [ ] purpose
- [ ] int.rate
- [ ] installment
- [x] log.annual.inc
- [ ] dti
- [ ] fico
- [x] days.with.cr.line
- [ ] revol.bal
- [x] revol.util
- [x] inq.last.6mths
- [x] delinq.2yrs
- [x] pub.rec
- [ ] not.fully.paid

Explanation

From summary(loans), we can read that log.annual.inc, days.with.cr.line, revol.util, inq.last.6mths, delinq.2yrs and pub.rec are missing values.

Submit You have used 0 of 2 attempts

Answers are displayed within the problem

Problem 1.3 - Preparing the Dataset

1 point possible (graded)

Which of the following is the best reason to fill in the missing values for these variables instead of removing observations with missing data? (Hint: you can use the subset() function to build a data frame with the observations missing at least one value. To test if a variable, for example pub.rec, is missing a value, use is.na(pub.rec).)

Calculator

- If we remove the missing observations there will be too little remaining data, leading to overfitting in our models.
- We want to be able to predict risk for all borrowers, instead of just the ones with all data reported. ✔
- In this dataset the observations with missing data have a much different rate of not paying in full, so removing them would bias subsequent models.

Explanation
Answering this question requires analyzing the loans with missing data. We can build a data frame limited to observations with some missing data with the following command:
missing = subset(loans, is.na(log.annual.inc) | is.na(days.with.cr.line) | is.na(revol.util) | is.na(inq.last.6mths) | is.na(delinq.2yrs) | is.na(pub.rec))
From nrow(missing), we see that only 62 of 9578 loans have missing data; removing this small number of observations would not lead to overfitting. From table(missing$not.fully.paid), we see that 12 of 62 loans with missing data were not fully paid, or 19.35%. This rate is similar to the 16.01% across all loans, so the form of biasing described is not an issue. However, to predict risk for loans with missing data we need to fill in the missing values instead of removing the observations.

Submit You have used 0 of 1 attempt

Answers are displayed within the problem

Problem 1.4 - Preparing the Dataset

1 point possible (graded)
For the rest of this problem, we'll be using a revised version of the dataset that has the missing values filled in with multiple imputation (which was discussed in the Recitation of this Unit). To ensure everybody has the same data frame going forward, you can either run the commands below in your R console (if you haven't already, run the command install.packages("mice") first), or you can download and load into R the dataset we created after running the imputation: loans_imputed.csv.

IMPORTANT NOTE: On certain operating systems, the imputation results are not the same even if you set the random seed. If you decide to do the imputation yourself, please still read the provided imputed dataset (loans_imputed.csv) into R and compare your results, using the summary function. If the results are different, please make sure to use the data in loans_imputed.csv for the rest of the problem.

```
library(mice)

set.seed(144)

vars.for.imputation = setdiff(names(loans), "not.fully.paid")

imputed = complete(mice(loans[vars.for.imputation]))

loans[vars.for.imputation] = imputed
```

Note that to do this imputation, we set vars.for.imputation to all variables in the data frame except for not.fully.paid, to impute the values using all of the other independent variables.

What best describes the process we just used to handle missing values?

- We removed all observations with a missing value.
- We filled each missing value with the average of all other values for that variable.
- We predicted missing variable values using the available independent variables for each observatio ✔

Calculator

- [x] We predicted missing variable values using the available independent and dependent variables for each observation.

Explanation
Imputation predicts missing variable values for a given observation using the variable values that are reported. We called the imputation on a data frame with the dependent variable not.fully.paid removed, so we predicted the missing values using only other independent variables.

Submit You have used 0 of 1 attempt

Answers are displayed within the problem

Problem 2.1 - Prediction Models

1 point possible (graded)
Now that we have prepared the dataset, we need to split it into a training and testing set. To ensure everybody obtains the same split, **set the random seed to 144 (even though you already did so earlier in the problem)** and use the sample.split function to select the 70% of observations for the training set (the dependent variable for sample.split is not.fully.paid). Name the data frames train and test.

Now, use logistic regression trained on the training set to predict the dependent variable not.fully.paid using all the independent variables.

Which independent variables are significant in our model? (Significant variables have at least one star, or a Pr(>|z|) value less than 0.05.) Select all that apply.

- [x] credit.policy
- [x] purpose2 (credit card)
- [x] purpose3 (debt consolidation)
- [ ] purpose4 (educational)
- [ ] purpose5 (home improvement)
- [x] purpose6 (major purchase)
- [x] purpose7 (small business)
- [ ] int.rate
- [x] installment
- [x] log.annual.inc
- [ ] dti
- [ ] fico

Calculator

- [x] fico
- [ ] days.with.cr.line
- [x] revol.bal
- [ ] revol.util
- [x] inq.last.6mths
- [ ] delinq.2yrs
- [x] pub.rec

Explanation
The following steps are needed to split the data:
library(caTools)
set.seed(144)
spl = sample.split(loans$not.fully.paid, 0.7)
train = subset(loans, spl == TRUE)
test = subset(loans, spl == FALSE)
The model can be trained and summarized with the following commands:
mod = glm(not.fully.paid~., data=train, family="binomial")
summary(mod)
Variables that are significant have at least one star in the coefficients table of the summary output. Note that some have a positive coefficient (meaning that higher values of the variable lead to an increased risk of defaulting) and some have a negative coefficient (meaning that higher values of the variable lead to a decreased risk of defaulting).

Submit You have used 0 of 3 attempts

Answers are displayed within the problem

Problem 2.2 - Prediction Models

0.0/4.0 points (graded)
Consider two loan applications, which are identical other than the fact that the borrower in Application A has FICO credit score 700 while the borrower in Application B has FICO credit score 710.

Let Logit(A) be the log odds of loan A not being paid back in full, according to our logistic regression model, and define Logit(B) similarly for loan B. What is the value of Logit(A) - Logit(B)?

Answer: 0.09317

Explanation
Because Application A is identical to Application B other than having a FICO score 10 lower, its predicted log odds differ by -0.009317 * -10 = 0.09317 from the predicted log odds of Application B.

Now, let O(A) be the odds of loan A not being paid back in full, according to our logistic regression model, and define O(B) similarly for loan B. What is the value of O(A)/O(B)? (HINT: Use the mathematical rule that exp(A + B + C) = exp(A)*exp(B)*exp(C). Also, remember that exp() is the exponential function in R.)

Answer: 1.0976

Calculator

Explanation
Using the answer from the previous question, the predicted odds of loan A not being paid back in full are exp(0.09317) = 1.0976 times larger than the predicted odds for loan B. Intuitively, it makes sense that loan A should have higher odds of non-payment than loan B, since the borrower has a worse credit score.

Submit You have used 0 of 7 attempts

Answers are displayed within the problem

Problem 2.3 - Prediction Models

0.0/4.0 points (graded)
Predict the probability of the test set loans not being paid back in full (remember type="response" for the predict function). **Store these predicted probabilities in a variable named predicted.risk and add it to your test set** (we will use this variable in later parts of the problem). Compute the confusion matrix using a threshold of 0.5.

What is the accuracy of the logistic regression model? Input the accuracy as a number between 0 and 1.

Answer: 0.8364

What is the accuracy of the baseline model? Input the accuracy as a number between 0 and 1.

Answer: 0.8399

Explanation
The confusion matrix can be computed with the following commands:
test$predicted.risk = predict(mod, newdata=test, type="response")
table(test$not.fully.paid, test$predicted.risk > 0.5)
2403 predictions are correct (accuracy 2403/2873=0.8364), while 2413 predictions would be correct in the baseline model of guessing every loan would be paid back in full (accuracy 2413/2873=0.8399).

Submit You have used 0 of 7 attempts

Answers are displayed within the problem

Problem 2.4 - Prediction Models

0.0/2.0 points (graded)
Use the ROCR package to compute the test set AUC.

Answer: 0.672

Explanation
The test set AUC can be computed with the following commands:
library(ROCR)
pred = prediction(test$predicted.risk, test$not.fully.paid)
as.numeric(performance(pred, "auc")@y.values)

The model has poor accuracy at the threshold 0.5. But despite the poor accuracy, we will see later how an investor can still leverage this logistic regression model to make profitable investments.

Submit You have used 0 of 5 attempts

Answers are displayed within the problem

Calculator

Answers are displayed within the problem

Problem 3.1 - A "Smart Baseline"

1 point possible (graded)

In the previous problem, we built a logistic regression model that has an AUC significantly higher than the AUC of 0.5 that would be obtained by randomly ordering observations.

However, LendingClub.com assigns the interest rate to a loan based on their estimate of that loan's risk. This variable, int.rate, is an independent variable in our dataset. In this part, we will investigate using the loan's interest rate as a "smart baseline" to order the loans according to risk.

Using the training set, build a bivariate logistic regression model (aka a logistic regression model with a single independent variable) that predicts the dependent variable not.fully.paid using only the variable int.rate.

The variable int.rate is highly significant in the bivariate model, but it is not significant at the 0.05 level in the model trained with all the independent variables. What is the most likely explanation for this difference?

- ✔ int.rate is correlated with other risk-related variables, and therefore does not incrementally improve the model when those other variables are included.
- This effect is likely due to the training/testing set split we used. In other splits, we could see the opposite effect.
- These models are trained on a different set of observations, so the coefficients are not comparable.

Explanation
To train the bivariate model, run the following command:
bivariate = glm(not.fully.paid~int.rate, data=train, family="binomial")
summary(bivariate)
Decreased significance between a bivariate and multivariate model is typically due to correlation. From cor(train$int.rate, train$fico), we can see that the interest rate is moderately well correlated with a borrower's credit score.
Training/testing set split rarely has a large effect on the significance of variables (this can be verified in this case by trying out a few other training/testing splits), and the models were trained on the same observations.

Submit You have used 0 of 1 attempt

Answers are displayed within the problem

Problem 3.2 - A "Smart Baseline"

2 points possible (graded)

Make test set predictions for the bivariate model. What is the highest predicted probability of a loan not being paid in full on the testing set?

Answer: 0.4266

With a logistic regression cutoff of 0.5, how many loans would be predicted as not being paid in full on the testing set?

Answer: 0

Explanation
Make and summarize the test set predictions with the following code:
pred.bivariate = predict(bivariate, newdata=test, type="response")
summary(pred.bivariate)

Calculator

According to the summary function, the maximum predicted probability of the loan not being paid back is 0.4266, which means no loans would be flagged at a logistic regression cutoff of 0.5.

Submit You have used 0 of 5 attempts

Answers are displayed within the problem

Problem 3.3 - A "Smart Baseline"

1 point possible (graded)
What is the test set AUC of the bivariate model?

Answer: 0.624

Explanation
The AUC can be computed with:
prediction.bivariate = prediction(pred.bivariate, test$not.fully.paid)
as.numeric(performance(prediction.bivariate, "auc")@y.values)

Submit You have used 0 of 5 attempts

Answers are displayed within the problem

Problem 4.1 - Computing the Profitability of an Investment

1 point possible (graded)
While thus far we have predicted if a loan will be paid back or not, an investor needs to identify loans that are expected to be profitable. If the loan is paid back in full, then the investor makes interest on the loan. However, if the loan is not paid back, the investor loses the money invested. Therefore, the investor should seek loans that best balance this risk and reward.

To compute interest revenue, consider a $c investment in a loan that has an annual interest rate r over a period of t years. Using continuous compounding of interest, this investment pays back c * exp(rt) dollars by the end of the t years, where exp(rt) is e raised to the r*t power.

How much does a $10 investment with an annual interest rate of 6% pay back after 3 years, using continuous compounding of interest? Hint: remember to convert the percentage to a proportion before doing the math. Enter the number of dollars, without the $ sign.

Answer: 11.97

Explanation
In this problem, we have c=10, r=0.06, and t=3. Using the formula above, the final value is 10*exp(0.06*3) = 11.97.

Submit You have used 0 of 3 attempts

Answers are displayed within the problem

Problem 4.2 - Computing the Profitability of an Investment

1 point possible (graded)
While the investment has value c * exp(rt) dollars after collecting interest, the investor had to pay $c for the investment. What is the profit to the investor if the investment is paid back in full?

Calculator

- c * exp(rt) - c ✔
- c * exp(rt)
- c * exp(rt) + c
- -c
- 0
- c

Explanation
A person's profit is what they get minus what they paid for it. In this case, the investor gets c * exp(rt) but paid c, yielding a profit of c * exp(rt) - c.

Submit You have used 0 of 2 attempts

Answers are displayed within the problem

Problem 4.3 - Computing the Profitability of an Investment

1 point possible (graded)
Now, consider the case where the investor made a $c investment, but it was not paid back in full. Assume, conservatively, that no money was received from the borrower (often a lender will receive some but not all of the value of the loan, making this a pessimistic assumption of how much is received). What is the profit to the investor in this scenario?

- c * exp(rt) - c
- c * exp(rt)
- c * exp(rt) + c
- -c ✔
- 0
- c

Explanation
A person's profit is what they get minus what they paid for it. In this case, the investor gets no money but paid c dollars, yielding a profit of -c dollars.

Submit You have used 0 of 2 attempts

Answers are displayed within the problem

Problem 5.1 - A Simple Investment Strategy

Calculator

0.0/2.0 points (graded)
In the previous subproblem, we concluded that an investor who invested c dollars in a loan with interest rate r for t years makes c * (exp(rt) - 1) dollars of profit if the loan is paid back in full and -c dollars of profit if the loan is not paid back in full (pessimistically).

In order to evaluate the quality of an investment strategy, we need to compute this profit for each loan in the test set. For this variable, we will assume a $1 investment (aka c=1). To create the variable, we first assign to the profit for a fully paid loan, exp(rt)-1, to every observation, and we then replace this value with -1 in the cases where the loan was not paid in full. All the loans in our dataset are 3-year loans, meaning t=3 in our calculations. Enter the following commands in your R console to create this new variable:

```
test$profit = exp(test$int.rate*3) - 1

test$profit[test$not.fully.paid == 1] = -1
```

What is the maximum profit of a $10 investment in any loan in the testing set (do not include the $ sign in your answer)?

Answer: 8.895

Explanation
From summary(test$profit), we see the maximum profit for a $1 investment in any loan is $0.8895. Therefore, the maximum profit of a $10 investment is 10 times as large, or $8.895.

Submit You have used 0 of 5 attempts

Answers are displayed within the problem

Problem 6.1 - An Investment Strategy Based on Risk

0.0/4.0 points (graded)
A simple investment strategy of equally investing in all the loans would yield profit $20.94 for a $100 investment. But this simple investment strategy does not leverage the prediction model we built earlier in this problem. As stated earlier, investors seek loans that balance reward with risk, in that they simultaneously have high interest rates and a low risk of not being paid back.

To meet this objective, we will analyze an investment strategy in which the investor only purchases loans with a high interest rate (a rate of at least 15%), but amongst these loans selects the ones with the lowest predicted risk of not being paid back in full. We will model an investor who invests $1 in each of the most promising 100 loans.

First, use the subset() function to build a data frame called highInterest consisting of the test set loans with an interest rate of at least 15%.

What is the average profit of a $1 investment in one of these high-interest loans (do not include the $ sign in your answer)?

Answer: 0.2251

What proportion of the high-interest loans were not paid back in full?

Answer: 0.2517

Explanation
The following two commands build the data frame highInterest and summarize the profit variable.
highInterest = subset(test, int.rate >= 0.15)
summary(highInterest$profit)
We read that the mean profit is $0.2251.
To obtain the breakdown of whether the loans were paid back in full, we can use
table(highInterest$not.fully.paid)

Calculator

110 of the 437 loans were not paid back in full, for a proportion of 0.2517.

Submit You have used 0 of 5 attempts

Answers are displayed within the problem

Problem 6.2 - An Investment Strategy Based on Risk

0.0/4.0 points (graded)

Next, we will determine the 100th smallest predicted probability of not paying in full by sorting the predicted risks in increasing order and selecting the 100th element of this sorted list. Find the highest predicted risk that we will include by typing the following command into your R console:

cutoff = sort(highInterest$predicted.risk, decreasing=FALSE)[100]

Use the subset() function to build a data frame called selectedLoans consisting of the high-interest loans with predicted risk not exceeding the cutoff we just computed. Check to make sure you have selected 100 loans for investment.

What is the profit of the investor, who invested $1 in each of these 100 loans (do not include the $ sign in your answer)?

Answer: 31.28

How many of 100 selected loans were not paid back in full?

Answer: 19

Explanation
selectedLoans can be constructed with the following code:
selectedLoans = subset(highInterest, predicted.risk <= cutoff)
You can check the number of elements with nrow(selectedLoans). The profit variable contains the profit for the $1 investment into each of the loans, so the following code computes the profit for all 100 loans:
sum(selectedLoans$profit)
The breakdown of whether each of the selected loans was fully paid can be computed with
table(selectedLoans$not.fully.paid)

We have now seen how analytics can be used to select a subset of the high-interest loans that were paid back at only a slightly lower rate than average, resulting in a significant increase in the profit from our investor's $100 investment. Although the logistic regression models developed in this problem did not have large AUC values, we see that they still provided the edge needed to improve the profitability of an investment portfolio.

We conclude with a note of warning. Throughout this analysis we assume that the loans we invest in will perform in the same way as the loans we used to train our model, even though our training set covers a relatively short period of time. If there is an economic shock like a large financial downturn, default rates might be significantly higher than those observed in the training set and we might end up losing money instead of profiting. Investors must pay careful attention to such risk when making investment decisions.

Submit You have used 0 of 5 attempts

Answers are displayed within the problem

Please remember not to ask for or post complete answers to homework questions in this discussion forum.

edX

Legal

Connect

Calculator