Kelly Xie
Prof. Tambe
Analytics in the Digital Economy
Apr 20, 2017

# Lab 5: Peer-to-peer lending

## 1. Data cleanup

```
# remove first row
train.loans = read.csv("~/Downloads/LoanStats3c.csv", skip=1)
test.loans = read.csv("~/Downloads/LoanStats3d.csv", skip=1)


# remove last 2 rows
train.loans = train.loans[-((nrow(train.loans)-1):nrow(train.loans)),]
test.loans = test.loans[-((nrow(test.loans)-1):nrow(test.loans)),]
```

## 2. Descriptive statistics

```
# create binary variable for grade
train.loans$highgrade = (train.loans$grade == "A") | (train.loans$grade == "B
")

# proportion of loans that received highgrade
percent.highgrade = mean(train.loans$highgrade)
print(percent.highgrade)

## [1] 0.4160905

# t-test results for differences in proportion of highgrade:
# whether the debtor is above or below the median income level
median.income = median(train.loans$annual_inc, na.rm = TRUE)
train.loans$annual_inc_above_median = train.loans$annual_inc >= median.income
t.test(highgrade~annual_inc_above_median, data = train.loans)

##
##  Welch Two Sample t-test
##
## data:  highgrade by annual_inc_above_median
## t = -45.554, df = 235520, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.09605877 -0.08813389
## sample estimates:
```

```
## mean in group FALSE  mean in group TRUE
##           0.3697967           0.4618931

# whether the loan request is above or below the median loan amount
median.loan = median(train.loans$loan_amnt, na.rm = TRUE)
train.loans$loan_above_median = train.loans$loan_amnt >= median.loan
t.test(highgrade~loan_above_median, data = train.loans)

##
##  Welch Two Sample t-test
##
## data:  highgrade by loan_above_median
## t = 32.046, df = 234900, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.06099924 0.06894691
## sample estimates:
## mean in group FALSE  mean in group TRUE
##           0.4491158           0.3841427

# whether the debtor rents their home or not
train.loans$home_rented = train.loans$home_ownership == "RENT"
t.test(highgrade~home_rented, data = train.loans)

##
##  Welch Two Sample t-test
##
## data:  highgrade by home_rented
## t = 14.688, df = 199440, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.02638399 0.03450975
## sample estimates:
## mean in group FALSE  mean in group TRUE
##           0.4280667           0.3976199
```

## 3. Logical classifier on training data

```
# perform a logistic regression that attempts to predict highgrade
# using annual income, home ownership, and loan amount as the predictors
model = glm(highgrade ~ annual_inc + home_ownership + loan_amnt,
            data = train.loans, family = binomial)

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(model)
```

Kelly Xie
Prof. Tambe
Analytics in the Digital Economy
Apr 20, 2017

```
##
## Call:
## glm(formula = highgrade ~ annual_inc + home_ownership + loan_amnt,
##     family = binomial, data = train.loans)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.4904  -1.0284  -0.8749   1.2716   1.7719
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             7.522e+00  2.666e+01   0.282    0.778
## annual_inc              7.281e-06  1.149e-07  63.376   <2e-16 ***
## home_ownershipMORTGAGE -7.699e+00  2.666e+01  -0.289    0.773
## home_ownershipOWN      -7.774e+00  2.666e+01  -0.292    0.771
## home_ownershipRENT     -7.853e+00  2.666e+01  -0.294    0.768
## loan_amnt              -4.331e-05  5.959e-07 -72.671   <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: 319984  on 235628  degrees of freedom
## Residual deviance: 312626  on 235623  degrees of freedom
## AIC: 312638
##
## Number of Fisher Scoring iterations: 6
```

```
# generate a vector of the probabilities that are predicted by the logistic r
egression
train.loans$predict_val = predict(model, type="response")

# create a new variable that classifies loans as being highgrade or not, base
d on predicted probabilities
train.loans$predict_highgrade = train.loans$predict_val > 0.47

# Evaluate how well this logistic regression-based classifier performs
# where accuracy is the proportion of rows in which the classifer prediction
is equal to its actual highgrade value.

# Benchmarks:
# 1. What is the accuracy of this classifer on the training data?
mean(train.loans$predict_highgrade == train.loans$highgrade)
```

```
## [1] 0.5958053
```

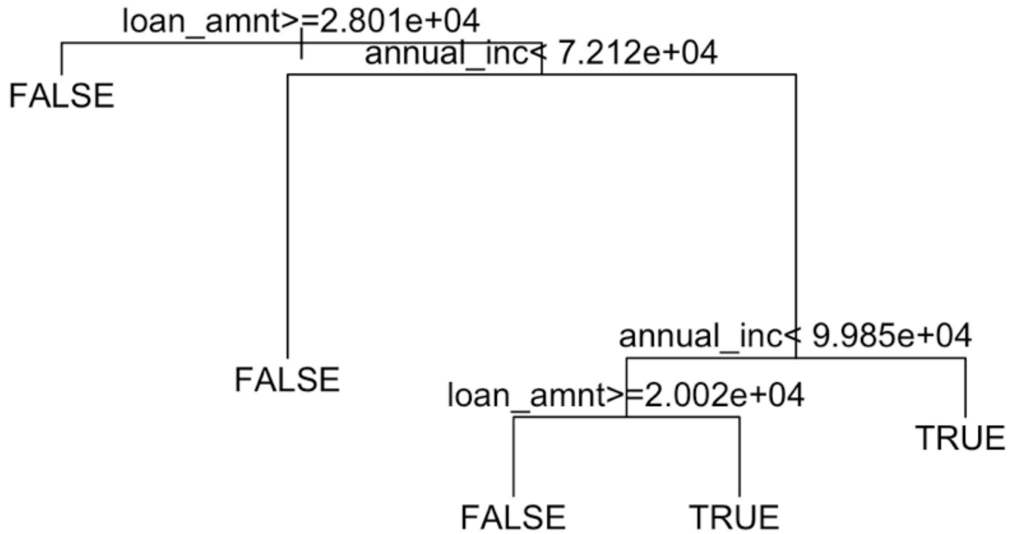
```
# 2. What would be the accuracy of a classifier that randomly assigns
# 0 and 1 values as the predicted class?
```

Kelly Xie
Prof. Tambe
Analytics in the Digital Economy
Apr 20, 2017

```
train.loans$predict_highgrade_random = runif(nrow(train.loans)) > 0.5 # assig
n random
mean(train.loans$predict_highgrade_random == train.loans$highgrade)

## [1] 0.4987926

# 3. What is the accuracy of a classifier that simply assigns a
# value of 0 to all rows for the predicted class?
train.loans$predict_highgrade_zero = 0 # assign 0s
mean(train.loans$predict_highgrade_zero == train.loans$highgrade)

## [1] 0.5839095
```

## 4. Supervised learning

```
library(rpart)
require(rpart)

# build a classification tree
fit = rpart(highgrade ~ annual_inc + home_ownership + loan_amnt,
            data = train.loans, method = "class")
plot(fit, margin=0.1)
text(fit)
```



Kelly Xie
Prof. Tambe
Analytics in the Digital Economy
Apr 20, 2017

```
# predict values using classification tree
train.loans$predict_highgrade_tree = predict(fit, type="class")

# get accuracy
mean(train.loans$predict_highgrade_tree == train.loans$highgrade)

## [1] 0.6093562
```

Machine learning based classifier is more accurate than logistic regression model in predicting whether loans are likely to have an A or B grade.

## 5. Model performance on test data

```
# Evaluate the accuracy of both of the classifiers on the test data.
test.loans$highgrade = (test.loans$grade == "A") | (test.loans$grade == "B")

# 1. logistic regression classifier
test.loans$predict_val = predict(model, newdata=test.loans, type="response")
test.loans$predict_highgrade_reg = test.loans$predict_val > 0.47 # probabilit
y threshold
mean(test.loans$predict_highgrade_reg == test.loans$highgrade)

## [1] 0.5871692

# 2. machine learning classifier
test.loans$predict_highgrade_tree = predict(fit, newdata=test.loans, type="cl
ass")
mean(test.loans$predict_highgrade_tree == test.loans$highgrade)

## [1] 0.6053076

# As a benchmark, what is the accuracy of a classifier that randomly
# assigns 0 and 1 values to the test data?
test.loans$predict_highgrade_random = runif(nrow(test.loans)) > 0.5 # assign
random
mean(test.loans$predict_highgrade_random == test.loans$highgrade)

## [1] 0.500298

# As another benchmark, what is the accuracy of a classifier that
# simply assigns a value of 0 to all rows of the test data?
test.loans$predict_highgrade_zero = 0 # assign 0s
mean(test.loans$predict_highgrade_zero == test.loans$highgrade)
```

```
## [1] 0.5465584
```