Lending Club R Notebook

Code

This is an R Markdown (http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Hide

```
#install.packages("rpart.plot")

library(TeachingDemos)
library(dplyr)
library(readr)
library(lubridate)
library(tidyverse)
library(magrittr)
library(rpart)
library(knitr)
library(rpart.plot)

setwd("~/Fall 2019/Oidd 245/Lab 4 - Lending Club")
```

Hide

```
trainingdata = read.csv(file = "training.csv", skip=1)
trainingdata = head(trainingdata, -2)
abcd = set.seed(9)
trainingdata$highgrade = trainingdata$grade == "A"|  trainingdata$grade == "B"
trainingdata$highincome = trainingdata$annual_inc >= median(trainingdata$annual_inc)
trainingdata$highloan = trainingdata$loan_amnt >= median(trainingdata$loan_amnt)
trainingdata$rent = trainingdata$home_ownership == "RENT"

t.test(highgrade ~ highincome, data = trainingdata)
```

```
    Welch Two Sample t-test

data:  highgrade by highincome
t = -45.554, df = 235518, 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
```

Hide

```
t.test(highgrade ~ highloan, data = trainingdata)
```

```
    Welch Two Sample t-test

data:  highgrade by highloan
t = 32.046, df = 234902, 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
```

Hide

```
t.test(highgrade ~ rent, data = trainingdata)
```

```
    Welch Two Sample t-test

data:  highgrade by rent
t = 14.688, df = 199444, 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
```

#looks like the the proportion of high grade loans based on each variable has a statistically significant p-value

Hide

```
graderegression = glm(highgrade ~ annual_inc + home_ownership + loan_amnt + verification_status
+ purpose, data=trainingdata, family = binomial)
```

```
glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Hide

```
summary(graderegression)
```

```
Call:
glm(formula = highgrade ~ annual_inc + home_ownership + loan_amnt +
    verification_status + purpose, family = binomial, data = trainingdata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.4904  -0.9499  -0.7030   1.1244   2.6029

Coefficients:
                                  Estimate Std. Error
(Intercept)                       8.188e+00  2.666e+01
annual_inc                        8.547e-06  1.216e-07
home_ownershipMORTGAGE           -8.055e+00  2.666e+01
home_ownershipOWN                -8.071e+00  2.666e+01
home_ownershipRENT               -8.180e+00  2.666e+01
loan_amnt                        -3.895e-05  6.762e-07
verification_statusSource Verified -6.533e-01  1.090e-02
verification_statusVerified      -9.497e-01  1.245e-02
purposecredit_card                8.271e-01  4.978e-02
purposedebt_consolidation        -8.011e-02  4.925e-02
purposehome_improvement          -3.269e-01  5.256e-02
purposehouse                     -2.032e+00  1.385e-01
purposemajor_purchase            -1.265e-01  5.963e-02
purposemedical                   -1.177e+00  7.063e-02
purposemoving                    -2.159e+00  1.037e-01
purposeother                     -1.173e+00  5.481e-02
purposerenewable_energy          -2.306e+00  3.299e-01
purposesmall_business            -1.844e+00  8.677e-02
purposevacation                  -1.294e+00  8.797e-02
purposewedding                   -4.688e-01  7.629e-01
                                 z value Pr(>|z|)
(Intercept)                        0.307   0.7588
annual_inc                        70.261  < 2e-16 ***
home_ownershipMORTGAGE            -0.302   0.7626
home_ownershipOWN                 -0.303   0.7621
home_ownershipRENT                -0.307   0.7590
loan_amnt                        -57.601  < 2e-16 ***
verification_statusSource Verified -59.928  < 2e-16 ***
verification_statusVerified      -76.262  < 2e-16 ***
purposecredit_card                16.617  < 2e-16 ***
purposedebt_consolidation         -1.627   0.1038
purposehome_improvement           -6.219 5.02e-10 ***
purposehouse                     -14.673  < 2e-16 ***
purposemajor_purchase             -2.121   0.0339 *
purposemedical                   -16.659  < 2e-16 ***
purposemoving                    -20.814  < 2e-16 ***
purposeother                     -21.394  < 2e-16 ***
purposerenewable_energy           -6.990 2.74e-12 ***
purposesmall_business            -21.251  < 2e-16 ***
purposevacation                  -14.712  < 2e-16 ***
purposewedding                    -0.614   0.5389
---
```

```
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: 290586  on 235609  degrees of freedom
AIC: 290626

Number of Fisher Scoring iterations: 6
```

Hide

```
mean(trainingdata$highgrade == trainingdata$predictforgrade)
```

```
[1] 0.66241
```

Hide

```
mean(0 == trainingdata$highgrade)
```

```
[1] 0.5839095
```

Hide

```
mean(rbinom(n=nrow(trainingdata), size=1, prob = .5)== trainingdata$highgrade)
```

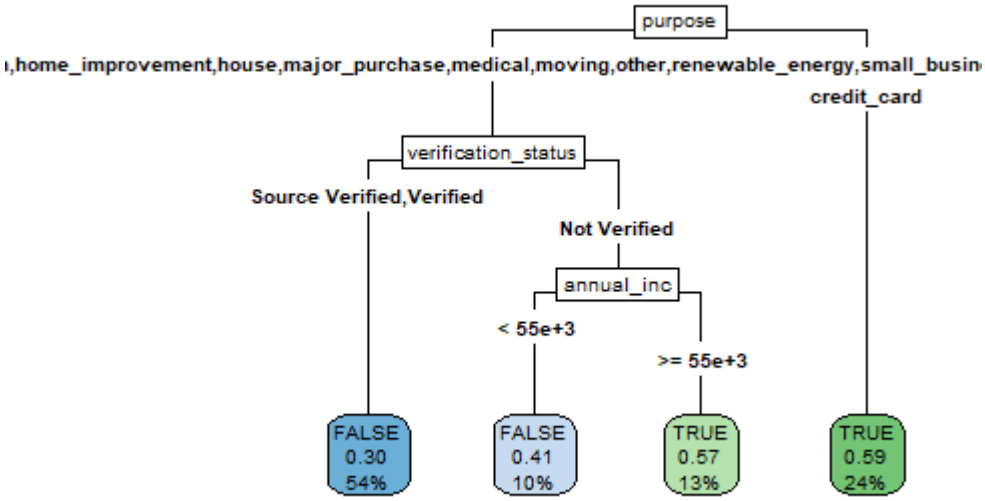
```
[1] 0.5000573
```

#the predict for grade regression does a better job than assinging 0 or randomly assigning numbers. It does so at 66% versus 58% or 50%

Hide

```
#4 classification tree

classtree = rpart(highgrade ~ annual_inc + home_ownership + loan_amnt + verification_status + pu
rpose, data=trainingdata, method = "class")
rpart.plot(classtree, type=5)
```



Hide

```
trainingdata$tryto = predict(classtree, type="class")

mean(trainingdata$highgrade == trainingdata$tryto)
```

```
[1] 0.6475858
```

Hide

```
mean(0 == trainingdata$highgrade)
```

```
[1] 0.5839095
```

Hide

```
mean(rbinom(n=nrow(trainingdata), size=1, prob = .5)== trainingdata$highgrade)
```

```
[1] 0.5010546
```

#the classtree is just a little bit less accurate than the regression, at 65%

Hide

```
testdata = head(testdata, -2) %>% filter(purpose != "educational")
testdata$highgrade = testdata$grade == "A"|  testdata$grade == "B"

testdata$predictit = predict(graderegression, newdata = testdata, type="response")
testdata$predictforgrade = testdata$predictit>= .5
testing = predict(classtree, testdata,  type="class")

mean(testdata$highgrade == testdata$predictforgrade)
```

```
[1] 0.6486046
```

Hide

```
mean(testdata$highgrade == testing)
```

```
[1] 0.6290075
```

Hide

```
mean(0 == testdata$highgrade)
```

```
[1] 0.5465632
```

Hide

```
mean(rbinom(n=nrow(testdata), size=1, prob = .5)== testdata$highgrade)
```

```
[1] 0.5001401
```

less accurate than with training data, but still better than assigning zero, or flipping a coin