Data Analytics and Machine Learning PS3

Huanyu Liu, Yong Jia Tan, Tongsu Peng, Sejal Bharati

4/22/2019

Question 1

a.

```
library(foreign)
library(data.table)
df = read.dta('LendingClub_LoanStats3a_v12.dta')
# (i)
df = df[df$loan_status == 'Fully Paid' | df$loan_status == 'Charged Off',]
df$default = 0
df[df$loan_status == 'Charged Off', 'default'] = 1
# (ii)
num_defaults = sum(df$default)
default_rate = num_defaults / nrow(df)
print(paste('average default rate in the sample is',default_rate))
```

```
## [1] "average default rate in the sample is 0.143534963970364"
```

b.

(i)

```
out = glm(formula = default ~ grade, family = 'binomial', data = df)
summary(out)
```

```
##
## Call:
## glm(formula = default ~ grade, family = "binomial", data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8827  -0.6077  -0.5053  -0.3511   2.3736
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.75542    0.04203  -65.56   <2e-16 ***
## gradeB       0.76143    0.05061   15.04   <2e-16 ***
## gradeC       1.15967    0.05153   22.50   <2e-16 ***
## gradeD       1.46001    0.05381   27.13   <2e-16 ***
## gradeE       1.69834    0.06030   28.17   <2e-16 ***
## gradeF       1.97319    0.07933   24.87   <2e-16 ***
## gradeG       2.01395    0.12800   15.73   <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: 32423  on 39411  degrees of freedom
## Residual deviance: 30914  on 39405  degrees of freedom
```

```
## AIC: 30928
##
## Number of Fisher Scoring iterations: 5
```

The larger the coefficients are, the higher probability there would be default.

(ii)

```
# (ii)
test_stat = out$null.deviance - out$deviance
k = out$df.null - out$df.residual
pval_chisq = 1 - pchisq(test_stat,df = k)
print(pval_chisq)
```

```
## [1] 0
```

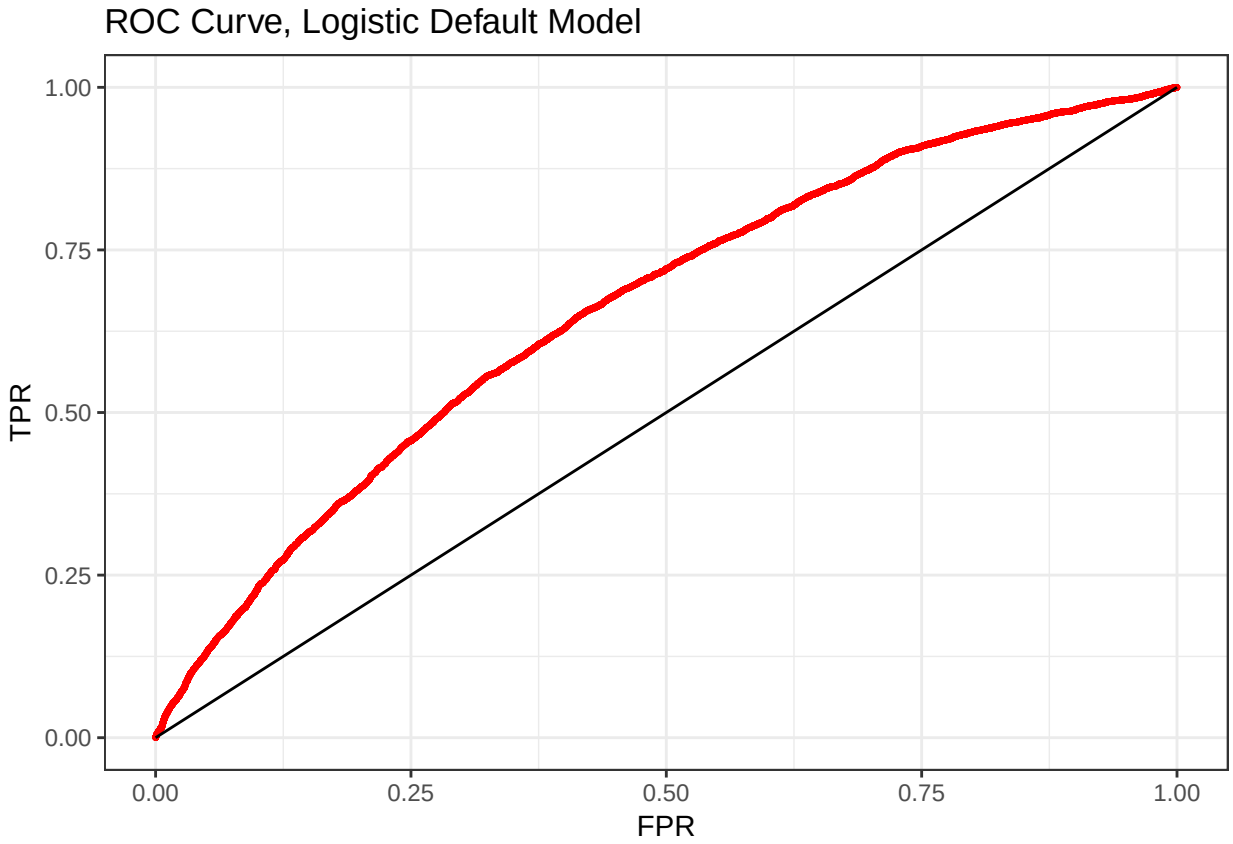
The p-value of chi-square test is 0, so that we can reject the null hypothesis, which mean at least one coefficient should not be 0. Therefore, this model performs better than the null model.

(iii)

```
# (iii)
phat = predict(out, type = 'response')
ranks = rank(phat, ties.method = 'first')
deciles = cut(ranks, quantile(ranks, probs=0:10/10), include.lowest=T)
deciles = as.numeric(deciles)
df2 = data.frame(deciles=deciles,phat=phat,default=df$default)
lift=aggregate(df2,by=list(deciles),FUN="mean",data=df2)
lift=lift[,c(2,4)]
lift[,3]=lift[,2]/mean(df$default)
names(lift)=c("decile","Mean Response","Lift Factor")
lift
```

```
##    decile Mean Response Lift Factor
## 1       1    0.06900051   0.4807226
## 2       2    0.05404720   0.3765438
## 3       3    0.09058615   0.6311086
## 4       4    0.11875159   0.8273356
## 5       5    0.11494545   0.8008184
## 6       6    0.14717077   1.0253304
## 7       7    0.17051510   1.1879691
## 8       8    0.19132200   1.3329296
## 9       9    0.20603908   1.4354626
## 10     10    0.27295789   1.9016822
```

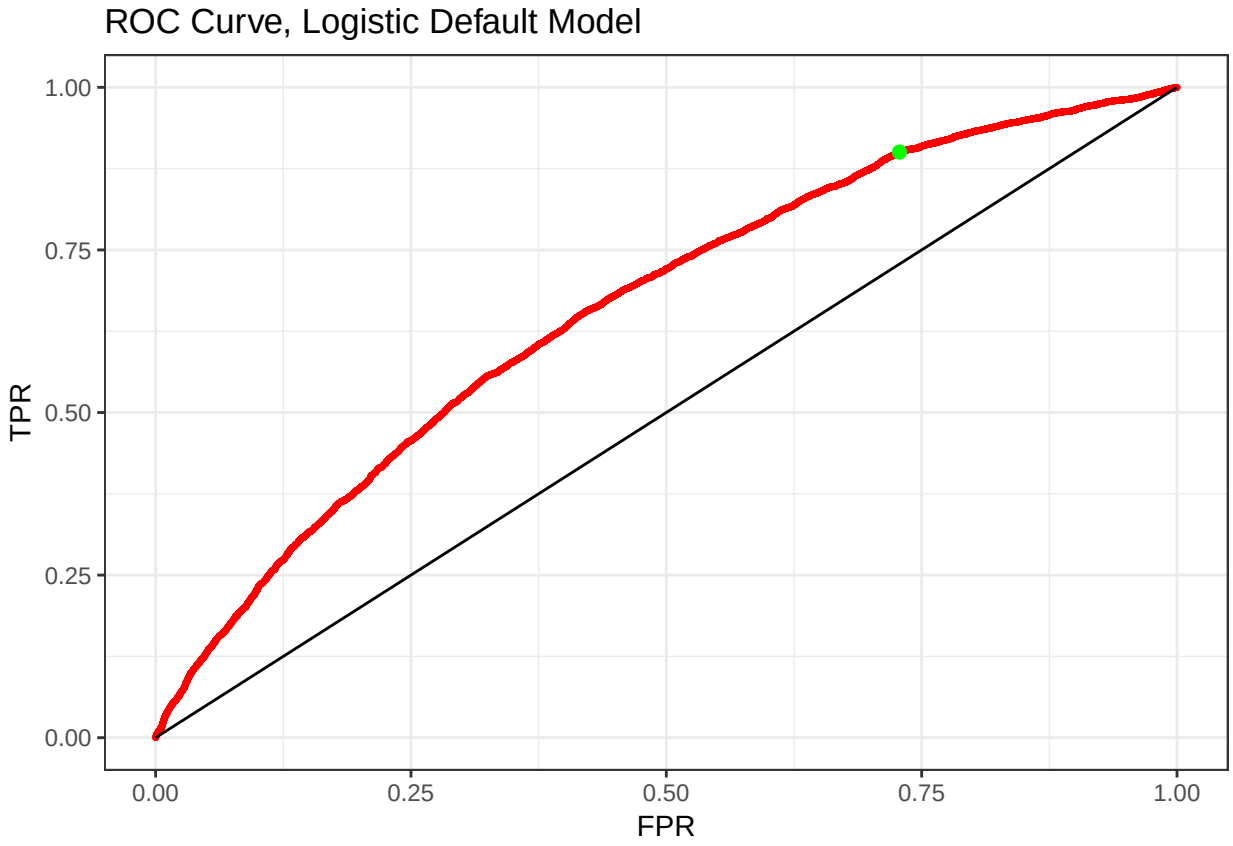
```
simple_roc <- function(labels, scores){
  labels <- labels[order(scores, decreasing=TRUE)]
  data.frame(TPR=cumsum(labels)/sum(labels), FPR=cumsum(!labels)/sum(!labels), labels)
}
roc = simple_roc(df$default == '1', phat)
TPR = roc$TPR
FPR = roc$FPR
library(ggplot2)
q = qplot(FPR,TPR,xlab = 'FPR', ylab = 'TPR', col = I('red'), main = 'ROC Curve, Logistic Default Model
q = q + geom_segment(aes(x = 0, xend = 1, y = 0, yend = 1)) + theme_bw()
q
```



The mean response of lift table should be higher with higher decile, same with lift factor.
The x-axis is the false positive rate, and the y-axis is the true positive rate. The line of ROC curve is tracing out the true and false positives for different cutoffs.
According to the result lift table and ROC, it is better than random guess.

(iv)

```
#(iv)
profits = sum(df$default == '0')*(1-FPR)*1
loss = sum(df$default == '1')*(1-TPR)*10
net = profits - loss
index = which.max(net)
cutoff = phat[index]
q + geom_point(aes(x = FPR[index], y = TPR[index], col = I('green'), size = I(2)))
```




```
cut_off_p = TPR[index]
print(paste('The cutoff probability should be',cut_off_p))
```

```
## [1] "The cutoff probability should be 0.900477284779919"
```

c.

(i)

```
#(i)
out2 = glm(formula = default ~ loan_amnt + annual_inc, family = 'binomial', data = df)
```

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

```
summary(out2)
```

```
##
## Call:
## glm(formula = default ~ loan_amnt + annual_inc, family = "binomial",
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8525  -0.5832  -0.5393  -0.4766   4.4804
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.725e+00  3.213e-02  -53.71   <2e-16 ***
## loan_amnt    3.484e-05  2.081e-06   16.74   <2e-16 ***
## annual_inc  -7.089e-06  4.663e-07  -15.20   <2e-16 ***
```

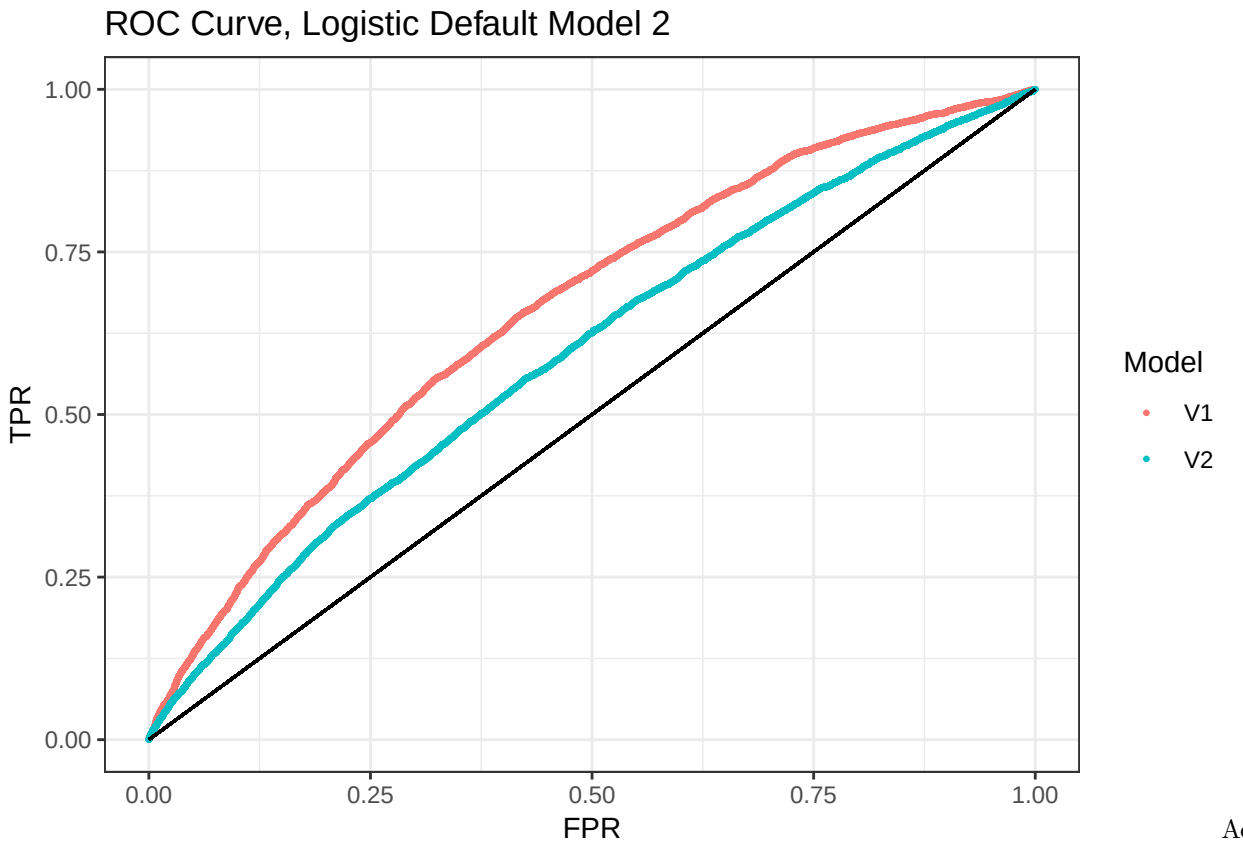
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 32423  on 39411  degrees of freedom
## Residual deviance: 32027  on 39409  degrees of freedom
## AIC: 32033
##
## Number of Fisher Scoring iterations: 5
```

```
phat2 = predict(out2, type = 'response')
deciles2 = cut(phat2, quantile(phat2, probs=0:10/10), include.lowest=T)
deciles2 = as.numeric(deciles2)
df3 = data.frame(deciles=deciles2,phat=phat2,default=df$default)
lift2=aggregate(df3,by=list(deciles2),FUN="mean",data=df2)
lift2=lift2[,c(2,4)]
lift2[,3]=lift2[,2]/mean(df$default)
names(lift2)=c("decile","Mean Response","Lift Factor")
lift2
```

```
##    decile Mean Response Lift Factor
## 1       1    0.08846641   0.6163405
## 2       2    0.10462164   0.7288930
## 3       3    0.11218030   0.7815538
## 4       4    0.12636664   0.8803893
## 5       5    0.13270743   0.9245652
## 6       6    0.14260340   0.9935098
## 7       7    0.15343647   1.0689832
## 8       8    0.14978421   1.0435381
## 9       9    0.20248668   1.4107133
## 10     10    0.22272958   1.5517444
```

```
roc2 = simple_roc(df$default == '1', phat2)
TPR2 = roc2$TPR
FPR2 = roc2$FPR
roc <- cbind(roc,Model = "V1")
roc2 <- cbind(roc2, Model = "V2")

New_ROC <- rbind(roc, roc2)
q = qplot(FPR,TPR, data = New_ROC, xlab = 'FPR', ylab = 'TPR', col = Model, main = 'ROC Curve, Logistic
q + geom_segment(aes(x = 0, xend = 1, y = 0, yend = 1), col = I('black')) + theme_bw()
```



According to the ROC curve, the area under grade model is larger than that of the alternative model, so the grade model is better.

(ii)

```
#(ii)
out3 = glm(formula = default ~ loan_amnt + annual_inc + term + int_rate, family = 'binomial', data = df)
```

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

```
summary(out3)
```

```
##
## Call:
## glm(formula = default ~ loan_amnt + annual_inc + term + int_rate,
##     family = "binomial", data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2520  -0.5868  -0.4694  -0.3598   4.1684
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.266e+00  6.055e-02 -53.942   <2e-16 ***
## loan_amnt       1.176e-06  2.311e-06   0.509    0.611
## annual_inc     -6.117e-06  4.643e-07 -13.173   <2e-16 ***
## term 60 months  4.538e-01  3.564e-02  12.732   <2e-16 ***
## int_rate        1.349e+01  4.560e-01  29.575   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

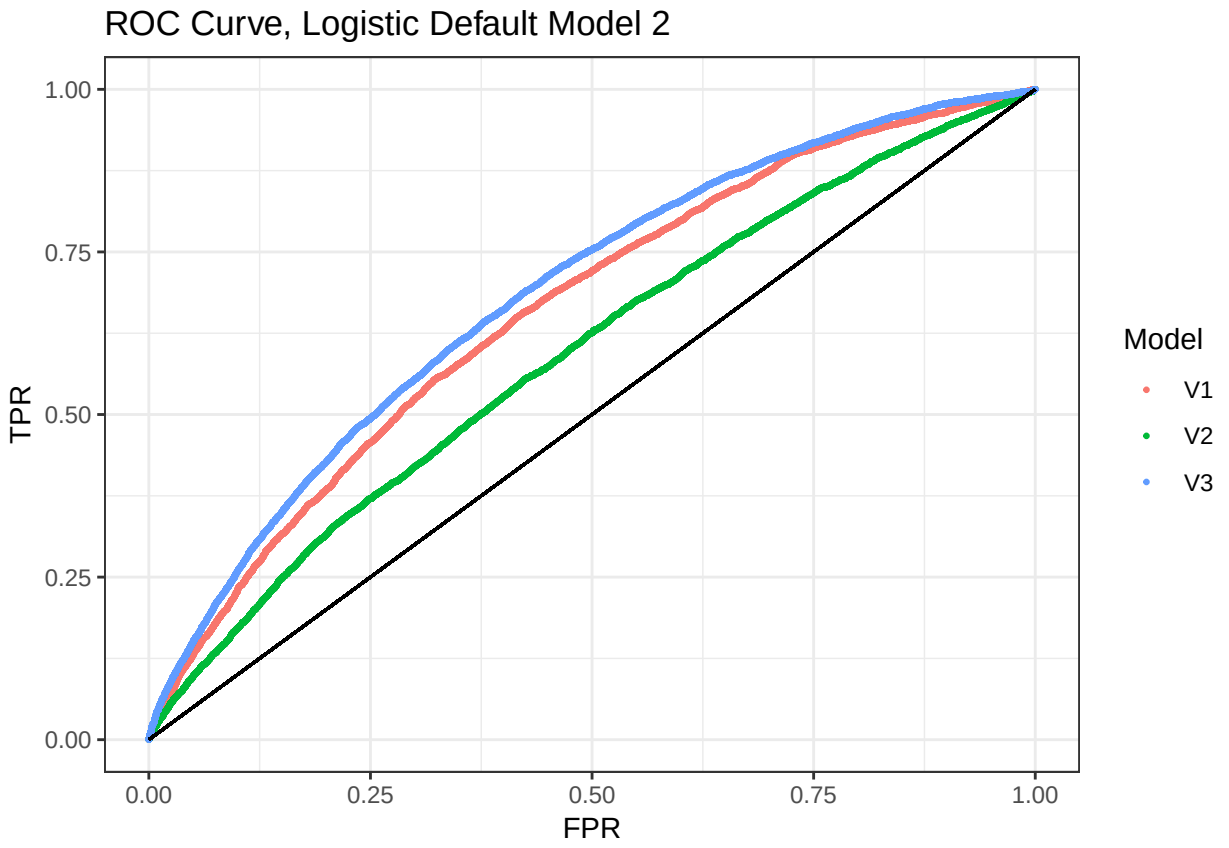
```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 32423  on 39411  degrees of freedom
## Residual deviance: 30418  on 39407  degrees of freedom
## AIC: 30428
##
## Number of Fisher Scoring iterations: 5
```

```
phat3 = predict(out3, type = 'response')
deciles3 = cut(phat3, quantile(phat3, probs=0:10/10), include.lowest=T)
deciles3 = as.numeric(deciles3)
df4 = data.frame(deciles=deciles3,phat=phat3,default=df$default)
lift3=aggregate(df4,by=list(deciles3),FUN="mean",data=df4)
lift3=lift3[,c(2,4)]
lift3[,3]=lift3[,2]/mean(df$default)
names(lift3)=c("decile","Mean Response","Lift Factor")
lift3
```

```
##    decile Mean Response Lift Factor
## 1       1    0.03652968   0.2545002
## 2       2    0.06368942   0.4437206
## 3       3    0.08043644   0.5603961
## 4       4    0.09895965   0.6894463
## 5       5    0.11646790   0.8114253
## 6       6    0.14514083   1.0111880
## 7       7    0.15782796   1.0995785
## 8       8    0.18751586   1.3064124
## 9       9    0.23572697   1.6422965
## 10     10    0.31303907   2.1809255
```

```
roc3 = simple_roc(df$default == '1', phat3)
TPR3 = roc3$TPR
FPR3 = roc3$FPR
roc3 <- cbind(roc3, Model = "V3")

New_ROC <- rbind(New_ROC, roc3)
q = qplot(FPR,TPR, data = New_ROC, xlab = 'FPR', ylab = 'TPR', col = Model, main = 'ROC Curve, Logistic
q + geom_segment(aes(x = 0, xend = 1, y = 0, yend = 1), col = I('black')) + theme_bw()
```



R treats term-variable as dummy variable.
The larger the coefficient means that variable affects more on the default probability. Positive coefficients means the smaller the variable the lower default probability, and the opposite way for the negative coefficients.
According to the ROC curve, this new model is better.
The default rate may be more relevant to the characteristics of the specific loan than to the grade of the borrowers.

(iii)

```
#(iii)
df$int_rate_sq = df$int_rate ^ 2
out4 = glm(formula = default ~ loan_amnt + annual_inc + term + int_rate + int_rate_sq, family = 'binomia
```

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

```
summary(out4)
```

```
##
## Call:
## glm(formula = default ~ loan_amnt + annual_inc + term + int_rate +
##     int_rate_sq, family = "binomial", data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0836  -0.5992  -0.4734  -0.3400   4.1124
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -4.035e+00  1.667e-01 -24.201  < 2e-16 ***
```

```
## loan_amnt       1.934e-06  2.307e-06   0.838    0.402
## annual_inc     -5.982e-06  4.635e-07 -12.905  < 2e-16 ***
## term 60 months  4.680e-01  3.548e-02  13.190  < 2e-16 ***
## int_rate        2.553e+01  2.458e+00  10.385  < 2e-16 ***
## int_rate_sq    -4.494e+01  8.985e+00  -5.002 5.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 32423  on 39411  degrees of freedom
## Residual deviance: 30393  on 39406  degrees of freedom
## AIC: 30405
##
## Number of Fisher Scoring iterations: 5
```

Yes, the coefficent on the square of interest rate is significant.
The positive coefficient on interest rate means that default rate will be lower with lower interest rate. While the negative coefficient on interest rate means that default rate will be higher with lower square of interest rate.
Because the square of interest rate is too small, overall interest rate will dominate its square. Therefore, default rate will be lower with lower interest rate in general.