Problem Set 3 Solution

Denis Mokanov

April 22, 2019

Question 1: Predicting default

a. We will use the column "loan_status" as the indicator for whether the loan was paid or there was a default.

(i) Drop all rows where "loan_status" is not equal to either "Fully Paid" or "Charged Off." Define the new variable Default as 1 (or TRUE) if "loan_status" is equal to "Charged Off", and 0 (or FALSE) otherwise.

```r
# housekeeping: remove all variables
rm(list = ls())
library(foreign)
library(data.table)
library(ggplot2)
# Download data and set as data.table
LendingClub_DT <- as.data.table(read.dta("LendingClub_LoanStats3a_v12.dta"))
# create status vector
status <- c("Fully Paid", "Charged Off")
LendingClub <- LendingClub_DT[loan_status %in% status]
# LendingClub[,Default:= loan_status== 'Charged Off']
LendingClub[, `:=`(Default, ifelse(loan_status == "Charged Off", 1, 0))]
```

(ii) Report the average default rate in the sample (number of defaults divided by total number of loans)

```r
(default_rate <- LendingClub[, mean(Default)])
```

```
## [1] 0.143535
```

b. LendingClub gives a "grade" to each borrower, designed as a score of each borrowers' creditworthiness. The best grade is "A", the worst grade is "G".

(i) Using the glm function, run a logistic regression of the Default variable on the grade. Report and explain the regression output, i.e., what is the interpretation of the coefficients? Do the numbers 'make sense'?

```r
out1 <- glm(Default ~ grade, family = "binomial", data = LendingClub)
summary(out1)
```

```
##
## Call:
## glm(formula = Default ~ grade, family = "binomial", data = LendingClub)
##
## 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
```

Interpretation: The estimated coefficient on `grade` is decreasing alphabetically, i.e. coefficient on `gradeB` is lower than the one for `gradeC`, the coefficient for `gradeC` is lower than the one on `gradeD` and so on. This makes sense: the higher the creditworthiness (`grade`) of the borrower the less likely he/she is to default.

(ii) Construct and report a test of whether the model performs better than the null model where only "beta0", and no conditioning information, is present in the logistic model.

As mentioned in class, the null hypothesis is a logistic regression with an intercept term and no slope. This is a model for which the intercept will be chosen to make the fitted probabilities that Y = 1 equal the frequency for which Y = 1 in the data. That is, the null model is simply to ignore X and compute the marginal probability that Y = 1 as opposed to the model which conditions on X.

```
(test_stat <- out1$null.deviance - out1$deviance)
```

```
## [1] 1508.097
```

```
# test stat is Chi-sq r.v. with k degrees of freedom

(df <- out1$df.null - out1$df.residual)
```

```
## [1] 6
```

```
(pvalue_chisq <- 1 - pchisq(test_stat, df = df))
```

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

```
# get the same test statistics using anova
anova(out1)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Default
##
## Terms added sequentially (first to last)
##
```

```
##
##       Df Deviance Resid. Df Resid. Dev
## NULL                  39411      32423
## grade  6   1508.1     39405      30914
```

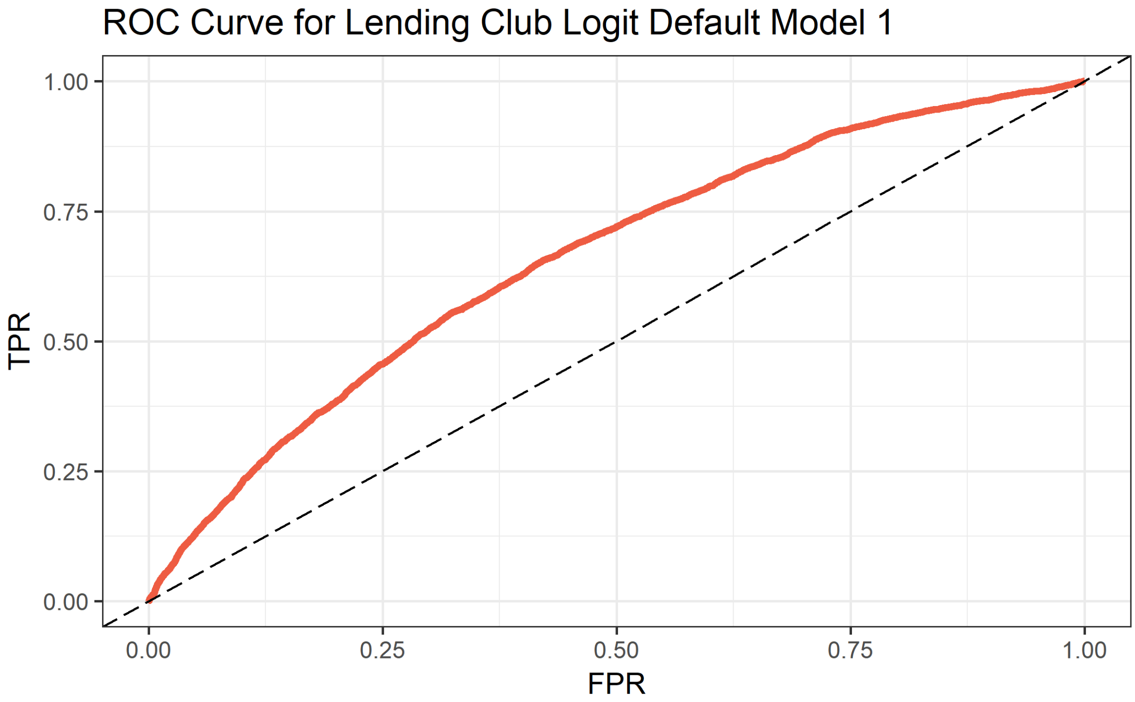
(iii) Construct the lift table and the ROC curve for this model. Explain the interpretation of the numbers in the lift table and the lines and axis in the ROC curve. Does the model perform better than a random guess?

```
# compute the 'lift' table
phat1 <- predict(out1, type = "response")

# use function ntile from dplyr to create deciles
library(dplyr)
deciles1 <- ntile(phat1, n = 10)
dt1 <- data.table(deciles = deciles1, phat = phat1, default = LendingClub$Default)
lift1 <- dt1[, lapply(.SD, mean), by = deciles]
lift1 <- lift1[, .(deciles, default)]
lift1[, `:=`(mean_response, default/mean(LendingClub$Default))]
setkey(lift1, deciles)
lift1
```

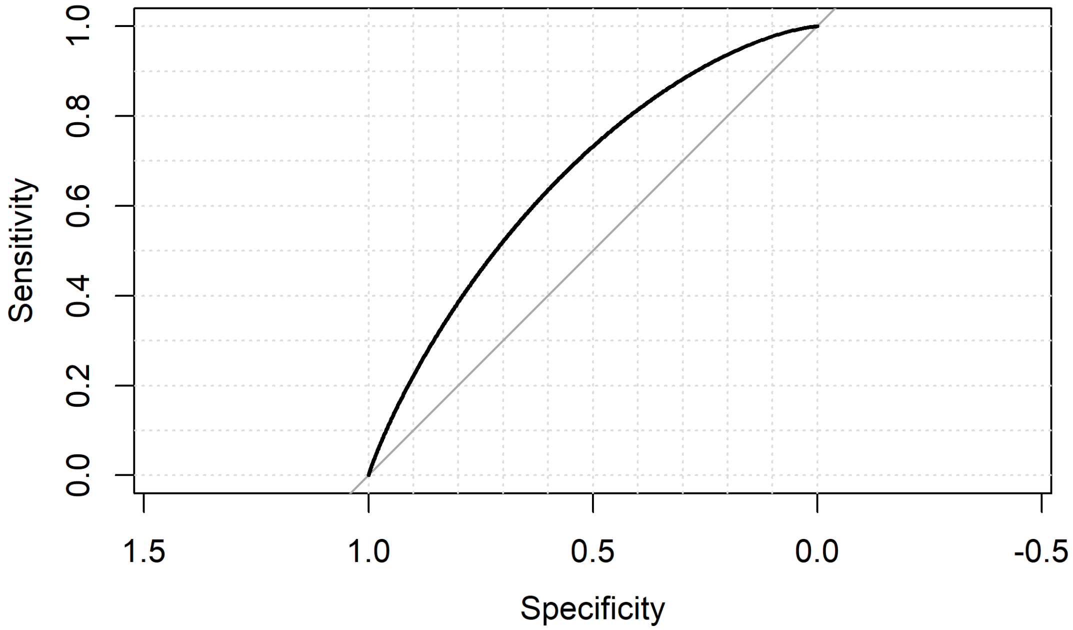
```
##     deciles    default mean_response
##  1:       1 0.06900051     0.4807226
##  2:       2 0.05404720     0.3765438
##  3:       3 0.09058615     0.6311086
##  4:       4 0.11875159     0.8273356
##  5:       5 0.11494545     0.8008184
##  6:       6 0.14738711     1.0268377
##  7:       7 0.17026135     1.1862013
##  8:       8 0.19132200     1.3329296
##  9:       9 0.20629282     1.4372304
## 10:      10 0.27277341     1.9003970
```

We see much higher default rates for higher deciles of fitted probability: 7% for decile 1 compared to 27% in decile 10. So we conclude that this model works well.

```
# compute a ROC curve define a function that creates the true and false
# positive rates
simple_roc <- function(labels, scores) {
    labels <- labels[order(scores, decreasing = TRUE)]
    data.frame(TPR = cumsum(labels)/sum(labels), FPR = cumsum(!labels)/sum(!labels),
        labels)
}

glm1_roc <- simple_roc(LendingClub$Default == "1", phat1)
TPR1 <- glm1_roc$TPR
FPR1 <- glm1_roc$FPR
data1 <- data.table(TPR = TPR1, FPR = FPR1)
# plot the corresponding ROC curve
ggplot(data1, aes(x = FPR, y = TPR)) + geom_line(color = "tomato2", size = 1.2) +
    ggtitle("ROC Curve for Lending Club Logit Default Model 1") + geom_abline(slope = 1,
    intercept = 0, linetype = "longdash") + theme_bw()
```



```
# can use pROC library in R to plot the ROC curve
library(pROC)
myROC <- roc(response = LendingClub$Default, predictor = phat1, smooth = TRUE,
    plot = TRUE, grid = TRUE)
```



```
# specificity = TN/(TN+FP) = TN/N; sensitivity = recall = TP/(TP+FN) = TP/P
```

From the ROC curve, we see this model performs better than the random guess (45-degree line) because the area under the ROC curve is larger.

(iv) Assume that each loan is for $100, and that you make a $1 profit if there is no default, but lose $10 if there is a default (both given in present value terms to keep things easy). Using data from the ROC curve (True Positive Rate and False Positive Rate) along with the average rate of default (total number of defaults divided by total number of loans), what is the cutoff default probability you should use as your decision criterion to maximize profits? Plot the corresponding point on the ROC curve.

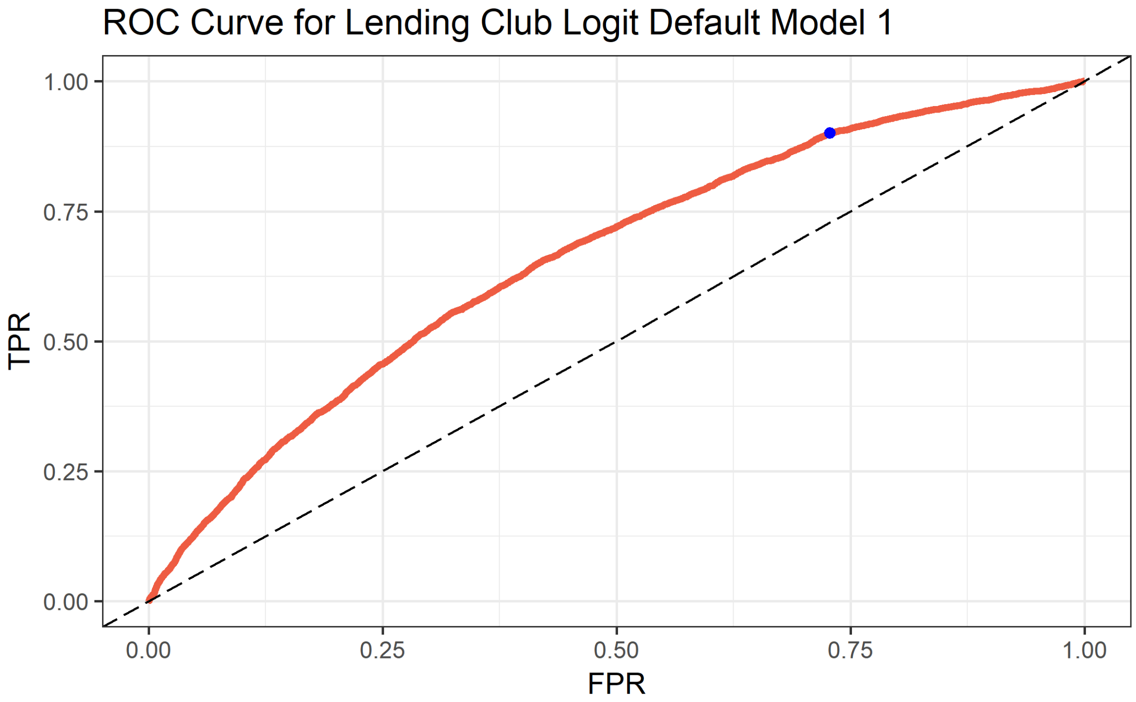
You are interested in maximizing 1 × (Number of people I gave a loan to that repaid) - 10 × (Number of people I gave a loan to that defaulted).

$$
\begin{aligned}
\max \quad & TN - 10FN \\
& (N - FP) - 10\,(P - TP) \\
& N\left(1 - \frac{FP}{N}\right) - 10P\left(1 - \frac{TP}{P}\right) \\
& N\,(1 - FPR) - 10P\,(1 - TPR)
\end{aligned}
$$

```r
# Calculate the maximum expected payoff
Data_iv = copy(data1)
Data_iv[, `:=`(Profit, 1)]
Data_iv[, `:=`(Default, -10)]
Data_iv[, `:=`(Cutoff_prob, Profit * (1 - FPR) * (1 - default_rate) + Default *
    (1 - TPR) * default_rate)]
Data_iv[, `:=`(Expected_Payoff, Cutoff_prob * nrow(LendingClub))]
Data_iv[, `:=`(Max_Payoff, max(Expected_Payoff))]
Data_iv[Expected_Payoff == Max_Payoff]
```

```
##          TPR       FPR Profit Default Cutoff_prob Expected_Payoff
## 1: 0.9004773 0.7283069      1     -10  0.08984573            3541
##    Max_Payoff
## 1:       3541
```

```r
# plot the corresponding ROC curve
ggplot(data1, aes(x = FPR, y = TPR)) + geom_line(color = "tomato2", size = 1.2) +
    ggtitle("ROC Curve for Lending Club Logit Default Model 1") + geom_abline(slope = 1,
    intercept = 0, linetype = "longdash") + geom_point(aes(x = Data_iv[Expected_Payoff ==
    Max_Payoff]$FPR, y = Data_iv[Expected_Payoff == Max_Payoff]$TPR), colour = "blue") +
    theme_bw()
```



c. Next, we will see if it is possible to do better than the internal "grade"-variable, using other information about the borrower and the loan as provided by LendingClub.

(i) First, consider a logistic regression model that uses only loan amount (`loan_amnt`) and annual income (`annual_inc`) as explanatory variables. Report the regression results. Show the lift table, comparing to the 'grade'-model from a. Plot the ROC curves of both the 'grade'-model and the alternative model. Which model performs better?

```
out2 <- glm(Default ~ loan_amnt + annual_inc, data = LendingClub, family = "binomial")
summary(out2)
```

```
##
## Call:
## glm(formula = Default ~ loan_amnt + annual_inc, family = "binomial",
##     data = LendingClub)
##
## 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)
##
```

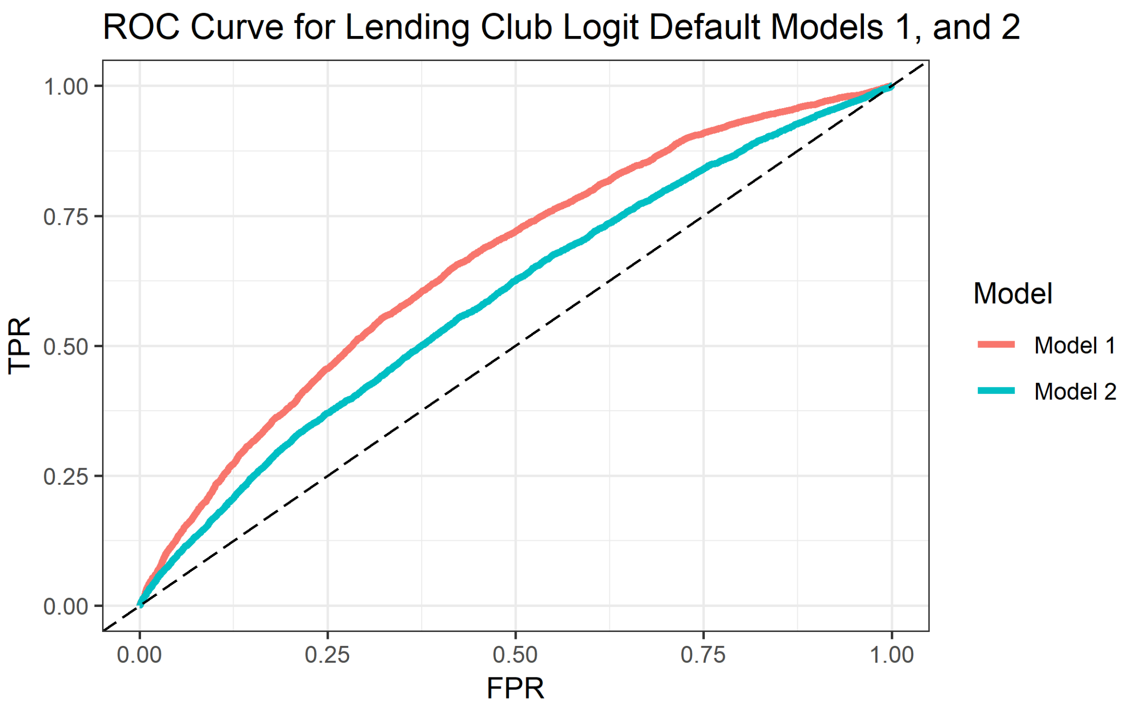
```
##     Null deviance: 32423  on 39411  degrees of freedom
## Residual deviance: 32027  on 39409  degrees of freedom
## AIC: 32033
##
## Number of Fisher Scoring iterations: 5
```

```r
# compute the 'lift' table
phat2 <- predict(out2, type = "response")
deciles2 <- ntile(phat2, 10)
dt2 <- data.table(deciles = deciles2, phat = phat2, default = LendingClub$Default)
lift2 <- dt2[, lapply(.SD, mean), by = deciles]
lift2 <- lift2[, .(deciles, default)]
lift2[, `:=`(mean_response, default/mean(LendingClub$Default))]
setkey(lift2, deciles)
lift2
```

```
##     deciles    default mean_response
##  1:       1 0.08853374     0.6168096
##  2:       2 0.10454199     0.7283382
##  3:       3 0.11215428     0.7813725
##  4:       4 0.12636387     0.8803699
##  5:       5 0.13270743     0.9245652
##  6:       6 0.14256722     0.9932578
##  7:       7 0.15351434     1.0695257
##  8:       8 0.14970820     1.0430086
##  9:       9 0.20248668     1.4107133
## 10:      10 0.22278609     1.5521382
```

Similarly, we see increasing default rate for higher deciles of the fitted probabilities (except from decile 7 to decile 8). Compared to the first model,

```r
glm2_roc <- simple_roc(LendingClub$Default == "1", phat2)
TPR2 <- glm2_roc$TPR
FPR2 <- glm2_roc$FPR
data2 <- data.table(TPR = TPR2, FPR = FPR2)
data1[, `:=`(Model, "Model 1")]
data2[, `:=`(Model, "Model 2")]
data <- rbind(data1, data2)
# plot the corresponding ROC curve
ggplot(data, aes(x = FPR, y = TPR, color = Model)) + geom_line(size = 1.2) +
    ggtitle("ROC Curve for Lending Club Logit Default Models 1, and 2") + geom_abline(slope = 1,
    intercept = 0, linetype = "longdash") + theme_bw()
```



From the ROC curves, the area under the curve (AUC) is larger for the "grade" model (Model 1). So Model 1 fits the data better.

(ii) Now, include also information from the loan itself. In particular, include the maturity of the loan (`term`) and the interest rate (`int_rate`) in the logistic regression. Report the output. How does `R` handle the term-variable? In particular, what is the interpretation of the regression coefficient? Again show the lift table and ROC curve relative to the original 'grade' model. Now, which model is better? What is the likely explanation for why this new model performs better/worse?

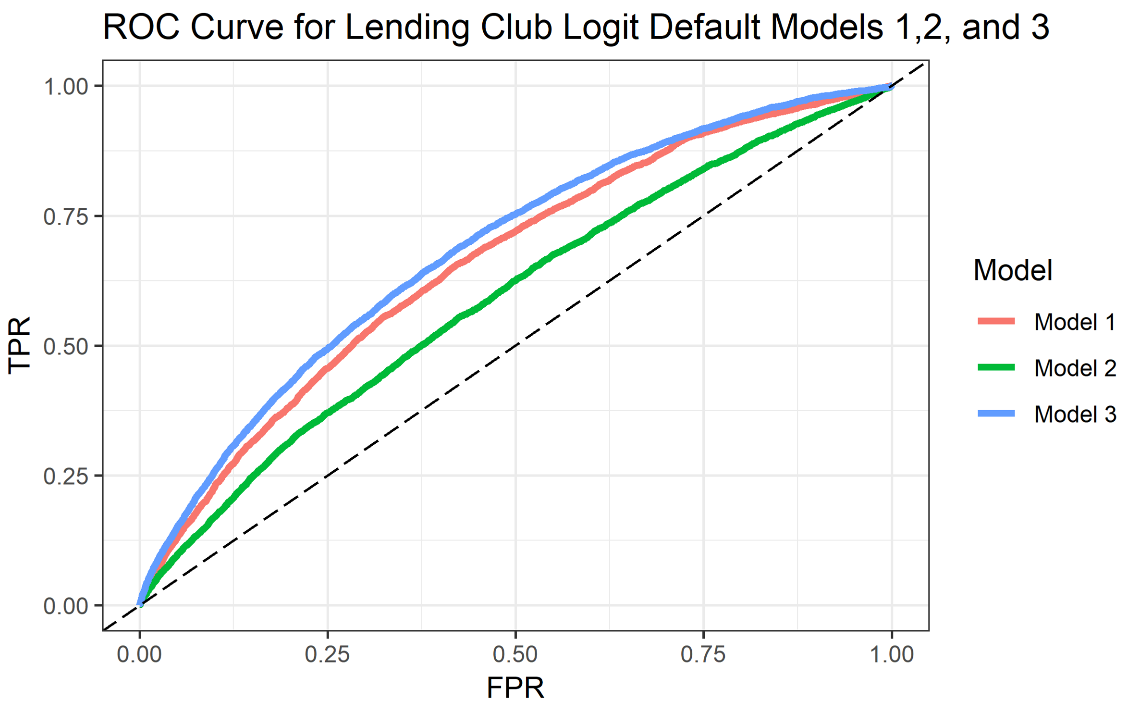
```r
out3 <- glm(Default ~ loan_amnt + annual_inc + term + int_rate, data = LendingClub,
    family = "binomial")
summary(out3)
```

```
##
## Call:
## glm(formula = Default ~ loan_amnt + annual_inc + term + int_rate,
##     family = "binomial", data = LendingClub)
##
## 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
```

```r
# compute the 'lift' table
phat3 <- predict(out3, type = "response")
deciles3 <- ntile(phat3, 10)
dt3 <- data.table(deciles = deciles3, phat = phat3, default = LendingClub$Default)
lift3 <- dt3[, lapply(.SD, mean), by = deciles]
lift3 <- lift3[, .(deciles, default)]
lift3[, `:=`(mean_response, default/mean(LendingClub$Default))]
setkey(lift3, deciles)
lift3
```

```
##     deciles    default mean_response
##  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 0.14510401     1.0109314
##  7:       7 0.15808171     1.1013463
##  8:       8 0.18726212     1.3046446
##  9:       9 0.23598072     1.6440643
## 10:      10 0.31286476     2.1797111
```

```r
glm3_roc <- simple_roc(LendingClub$Default == "1", phat3)
TPR3 <- glm3_roc$TPR
FPR3 <- glm3_roc$FPR
data3 <- data.table(TPR = TPR3, FPR = FPR3)
data3[, `:=`(Model, "Model 3")]
data <- rbind(data, data3)
# plot the corresponding ROC curve
ggplot(data, aes(x = FPR, y = TPR, color = Model)) + geom_line(size = 1.2) +
    ggtitle("ROC Curve for Lending Club Logit Default Models 1,2, and 3") +
    geom_abline(slope = 1, intercept = 0, linetype = "longdash") + theme_bw()
```



From the logistic regression output `out3`, we see that `R` created dummy variables for one of the two possible values of the `term` variable (60 months) and did not create a dummy for the other value (30 months).

- Interpretation of the coefficients
- `loan_amnt`: predicts default positively, the higher the loan amount the more likely is the individual to default,
- `annual_income`: predicts defaults negatively, the higher is the individual's income the less likely is the default,
- `term 60 months`: predicts default positively, the longer the loan term the more likely is the individual to default,
- `int_rate`: predicts default positively, the higher the interest rate on the loan the more likely is the individual to default,

From the ROC curve, we see this model performs better than the last two models since the area under the ROC curve is larger for Model 3.

Explanation for the outperformance of Model 3 relative to Model 1: in Model 1 the only independent variable is `grade` which corresponds to the creditworthiness of the individual (similar to a FICO score). However, there may be other factor determining the default of a loan that are not accounted for in the the assigned `grade`, such as the interest rate on the loan

(iii) Create the squared of the interest rate and add this variable to the last model. Is the coefficient on this variable significant? Please give an intuition for what the coefficients on both `int_rate` and its squared value imply for the relationship between defaults and the interest rate.

```r
# creat squared of interest rate variable
LendingClub[, `:=`(int_rate_sq, int_rate * int_rate)]
out4 <- glm(Default ~ loan_amnt + annual_inc + term + int_rate + int_rate_sq,
    data = LendingClub, family = "binomial")
summary(out4)
```

```
##
## Call:
## glm(formula = Default ~ loan_amnt + annual_inc + term + int_rate +
##     int_rate_sq, family = "binomial", data = LendingClub)
##
## 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
```

From the regression summary output, the coefficients on `int_rate` and `int_rate_sq` are both significant. The coefficient on `int_rate` is positive, reflecting the expected relation that higher default probabilities lead to high interest rates on the loan. The negative sign on the squared term means that the sensitivity of the default probability to `int_rate` is decreasing in the level of int_rate. In particular, one can rewrite these terms as follows:

$$\beta_1 \times int\_rate + \beta_2 \times int\_rate^2 = (\beta_1 + \beta_2 \times int\_rate) \times int\_rate.$$

Thus, the squared term captures a non-linearity in the relation between default rates and loan interest rates.