Predict Loan Offer Logistic Regression

Lucky

```
#### Personal Loan Offer using Logistic Regression
## Outcome Variable: Accept Bank Loan (0/1)
## Predictors: Demographic information and Bank information

# Set working directory
# Read CSV data
bank.df <- read.csv("UniversalBank.csv")
bank.df <- bank.df[ , -c(1, 5)]  # Drop ID and zip code columns.
# treat Education as categorical (R will create dummy variables)
bank.df$Education <- factor(bank.df$Education, levels = c(1, 2, 3),
                            labels = c("Undergrad", "Graduate",
"Advanced/Professional"))

# partition data
set.seed(2)
train.index <- sample(c(1:dim(bank.df)[1]), dim(bank.df)[1]*0.6)
train.df <- bank.df[train.index, ]
valid.df <- bank.df[-train.index, ]

# run logistic regression
# use glm() (general linear model) with family = "binomial" to fit a logistic
# regression.
logit.reg <- glm(PersonalLoan ~ ., data = train.df, family = "binomial")
options(scipen=999)
summary(logit.reg)

##
## Call:
## glm(formula = PersonalLoan ~ ., family = "binomial", data = train.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0417  -0.1861  -0.0641  -0.0182   4.1742
##
## Coefficients:
##                                 Estimate  Std. Error z value
## (Intercept)                  -15.5820754   2.2351307  -6.971
## Age                            0.0743198   0.0804261   0.924
## Experience                    -0.0593021   0.0801234  -0.740
## Income                         0.0627328   0.0040045  15.666
## Family                         0.5476224   0.0978787   5.595
## CCAvg                          0.1651545   0.0588724   2.805
## EducationGraduate              4.2286088   0.3614010  11.701
```

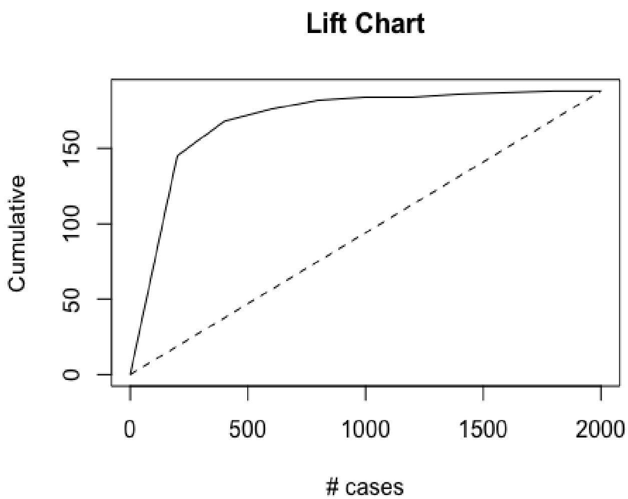
```
## EducationAdvanced/Professional   4.2208138   0.3622092  11.653
## Mortgage                         0.0011339   0.0007789   1.456
## SecuritiesAccount               -0.7064409   0.3820337  -1.849
## CDAccount                        3.5878387   0.4345389   8.257
## Online                          -0.5602502   0.2161742  -2.592
## CreditCard                      -1.2225669   0.2842447  -4.301
##                                           Pr(>|z|)
## (Intercept)                        0.00000000000314 ***
## Age                                         0.35545
## Experience                                  0.45922
## Income                          < 0.0000000000000002 ***
## Family                             0.00000002207361 ***
## CCAvg                                       0.00503 **
## EducationGraduate               < 0.0000000000000002 ***
## EducationAdvanced/Professional < 0.0000000000000002 ***
## Mortgage                                    0.14543
## SecuritiesAccount                           0.06443 .
## CDAccount                       < 0.0000000000000002 ***
## Online                                      0.00955 **
## CreditCard                         0.00001699463815 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1915.10  on 2999  degrees of freedom
## Residual deviance:  690.83  on 2987  degrees of freedom
## AIC: 716.83
##
## Number of Fisher Scoring iterations: 8
```

```
# use predict() with type = "response" to compute predicted probabilities.
logit.reg.pred <- predict(logit.reg, valid.df[, -8], type = "response")

# first 5 actual and predicted records
data.frame(actual = valid.df$PersonalLoan[1:5], predicted =
logit.reg.pred[1:5])
```

```
##   actual      predicted
## 1      0 0.000128278630
## 3      0 0.000005182088
## 4      0 0.117204971114
## 6      0 0.003390944766
## 7      0 0.023748906445
```

```
library(gains)
gain <- gains(valid.df$PersonalLoan, logit.reg.pred, groups=10)

# plot lift chart
# Different between Cummulative Personal Loan sorted using predicted values
```

```
and using average
plot(c(0,gain$cume.pct.of.total*sum(valid.df$PersonalLoan))~c(0,gain$cume.obs
),
     xlab="# cases", ylab="Cumulative", main="Lift Chart", type="l")
lines(c(0,sum(valid.df$PersonalLoan))~c(0, dim(valid.df)[1]), lty=2)
```




```
# compute deciles and plot decile-wise chart
heights <- gain$mean.resp/mean(valid.df$PersonalLoan)
midpoints <- barplot(heights, names.arg = gain$depth, ylim = c(0,9),
                     xlab = "Percentile", ylab = "Mean Response", main =
"Decile-wise lift chart")
# add labels to columns
text(midpoints, heights+0.5, labels=round(heights, 1), cex = 0.8)
```