Modeling techniques:

a.) Logical Regression Model.

Data | Explore | Test | Transform | Cluster | Associate | Model | Evaluate | Log

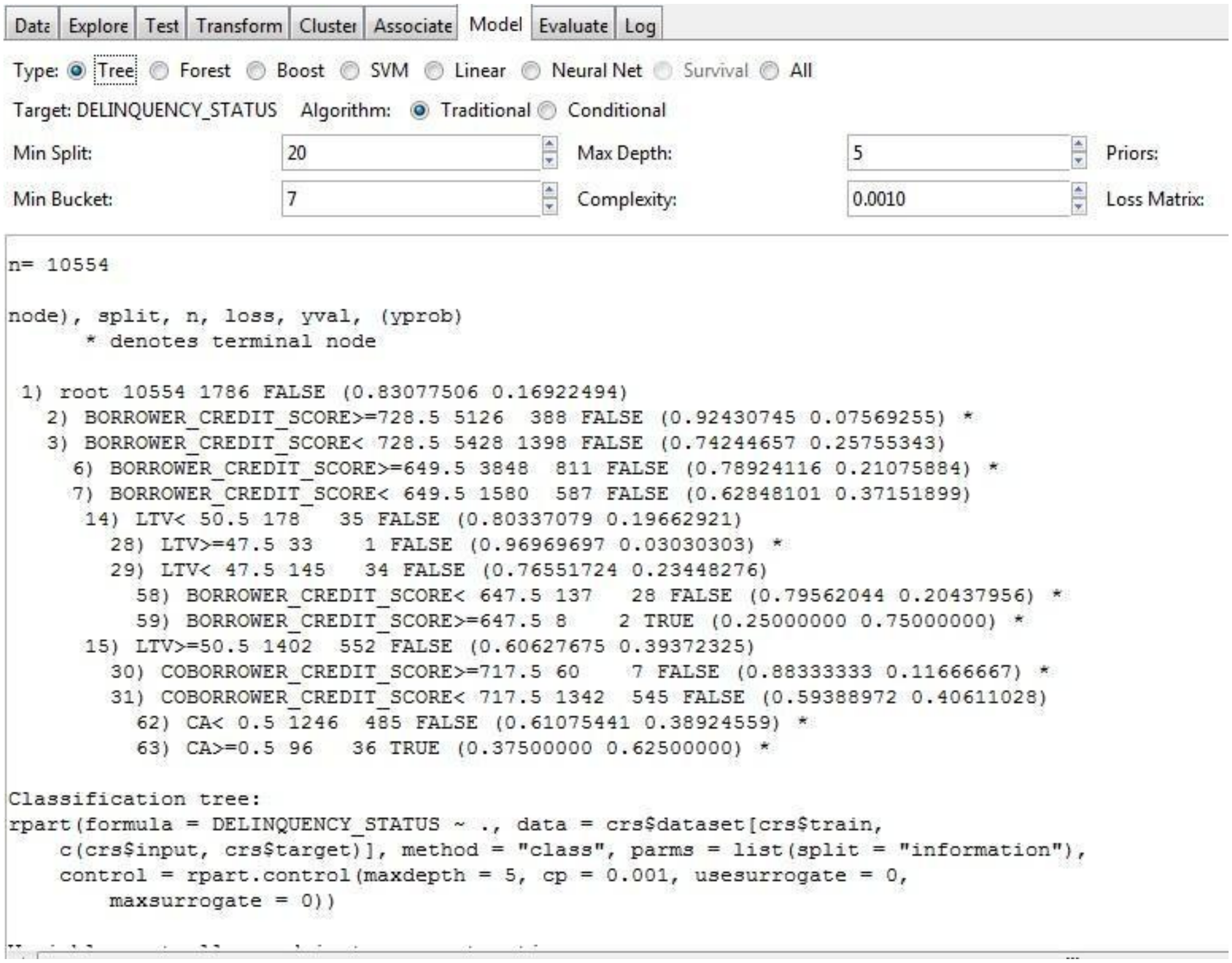
Type: Tree Forest Boost SVM Linear Neural Net Survival All

Numeric Generalized Poisson Logistic Probit Multinomial

Plot

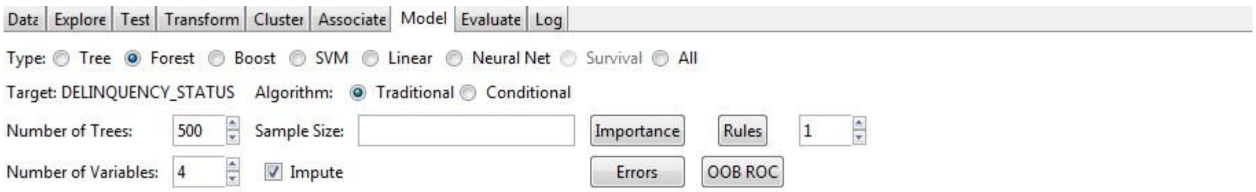
```
Summary of the Logistic Regression model (built using glm):

Call:
glm(formula = DELINQUENCY_STATUS ~ ., family = binomial(link = "logit"),
    data = crs$dataset[crs$train, c(crs$input, crs$target)])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.4229  -0.6291  -0.4326  -0.2536   3.3142

Coefficients:
                                    Estimate     Std. Error z value Pr(>|z|)
(Intercept)                    -0.11098544570  0.15794333367  -0.703    0.482
CHANNELC                       -0.14557731730  0.01943347052  -7.491 6.83e-14 ***
CHANNELR                       -0.22241734610  0.02049127125 -10.854  < 2e-16 ***
ORIGINAL_INTEREST_RATE          0.31982068799  0.01851968006  17.269  < 2e-16 ***
ORIGINAL_UNPAID_PRINCIPAL_BALANCE  0.00000096238  0.00000008024  11.994  < 2e-16 ***
LTV                             0.02310645439  0.00154262110  14.979  < 2e-16 ***
CLTV                            0.01555791631  0.00148280482  10.492  < 2e-16 ***
NUMBER_BORROWERS               -0.48392970162  0.01413597143 -34.234  < 2e-16 ***
DTI_RATIO                       0.01535661851  0.00055751897  27.545  < 2e-16 ***
BORROWER_CREDIT_SCORE          -0.00865178701  0.00011901166 -72.697  < 2e-16 ***
LOAN_PURPOSEP                  -0.84843724234  0.01848159053 -45.907  < 2e-16 ***
LOAN_PURPOSER                  -0.30805241307  0.01845252688 -16.694  < 2e-16 ***
AZ                              1.25094052235  0.03120143798  40.092  < 2e-16 ***
CA                              0.94717921103  0.02394018070  39.564  < 2e-16 ***
NV                              1.53258054433  0.05437159122  28.187  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

b.) Trees Model



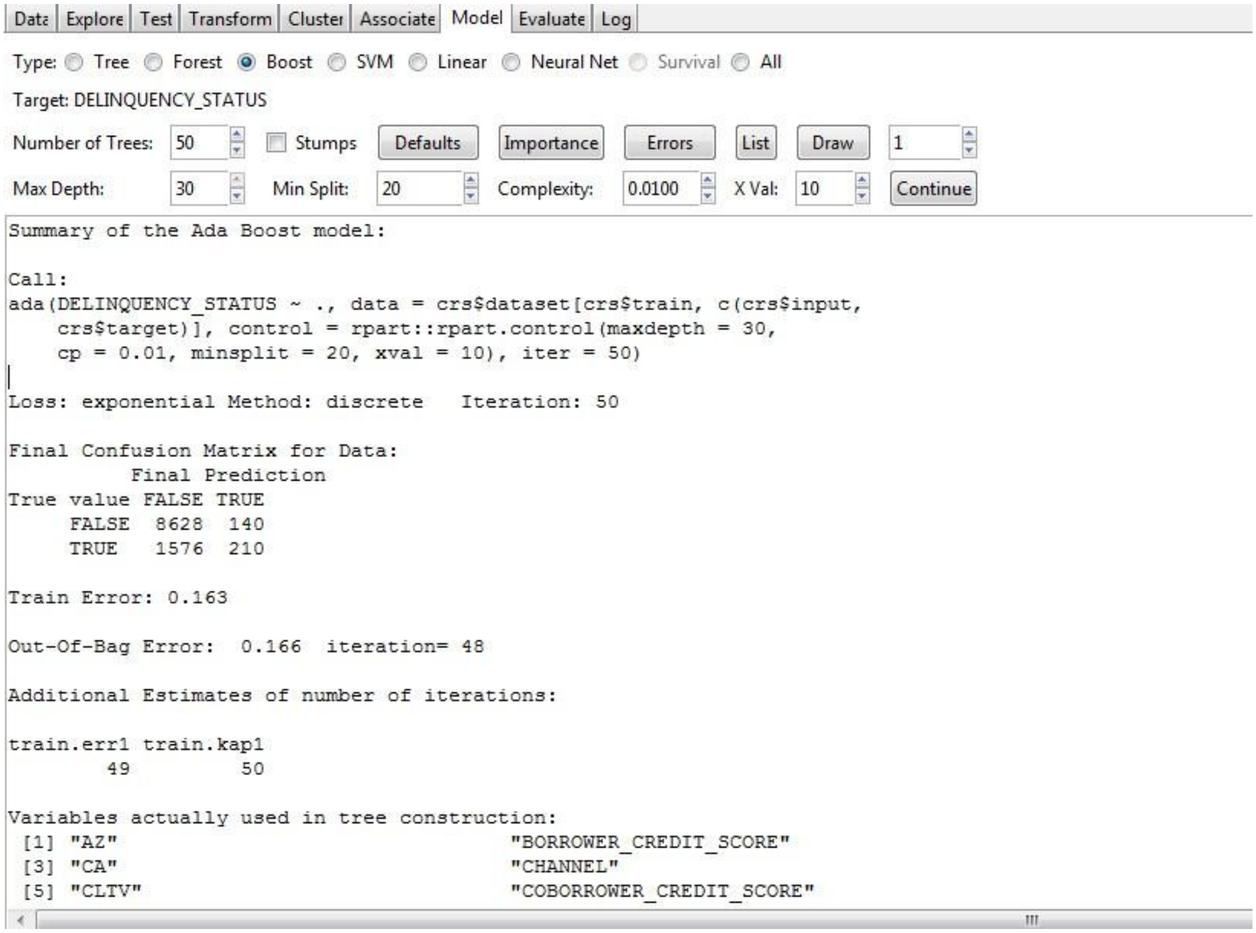

```
n= 10554

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 10554 1786 FALSE (0.83077506 0.16922494)
   2) BORROWER_CREDIT_SCORE>=728.5 5126  388 FALSE (0.92430745 0.07569255) *
   3) BORROWER_CREDIT_SCORE< 728.5 5428 1398 FALSE (0.74244657 0.25755343)
     6) BORROWER_CREDIT_SCORE>=649.5 3848  811 FALSE (0.78924116 0.21075884) *
     7) BORROWER_CREDIT_SCORE< 649.5 1580  587 FALSE (0.62848101 0.37151899)
      14) LTV< 50.5 178   35 FALSE (0.80337079 0.19662921)
        28) LTV>=47.5 33    1 FALSE (0.96969697 0.03030303) *
        29) LTV< 47.5 145   34 FALSE (0.76551724 0.23448276)
          58) BORROWER_CREDIT_SCORE< 647.5 137   28 FALSE (0.79562044 0.20437956) *
          59) BORROWER_CREDIT_SCORE>=647.5 8    2 TRUE (0.25000000 0.75000000) *
      15) LTV>=50.5 1402  552 FALSE (0.60627675 0.39372325)
        30) COBORROWER_CREDIT_SCORE>=717.5 60    7 FALSE (0.88333333 0.11666667) *
        31) COBORROWER_CREDIT_SCORE< 717.5 1342  545 FALSE (0.59388972 0.40611028)
          62) CA< 0.5 1246  485 FALSE (0.61075441 0.38924559) *
          63) CA>=0.5 96   36 TRUE (0.37500000 0.62500000) *

Classification tree:
rpart(formula = DELINQUENCY_STATUS ~ ., data = crs$dataset[crs$train,
    c(crs$input, crs$target)], method = "class", parms = list(split = "information"),
    control = rpart.control(maxdepth = 5, cp = 0.001, usesurrogate = 0,
        maxsurrogate = 0))
```

c.) Forest Trees Model




```
Summary of the Random Forest Model
==================================

Number of observations used to build the model: 10554
Missing value imputation is active.

Call:
 randomForest(formula = as.factor(DELINQUENCY_STATUS) ~ .,
              data = crs$dataset[crs$sample, c(crs$input, crs$target)],
              ntree = 500, mtry = 4, importance = TRUE, replace = FALSE, na.action = randomForest::na.roughfix)

               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 4

        OOB estimate of  error rate: 16.81%
Confusion matrix:
      FALSE TRUE class.error
FALSE  8588  180   0.0205292
TRUE   1594  192   0.8924972

Analysis of the Area Under the Curve (AUC)
==========================================

Call:
roc.default(response = crs$rf$y, predictor = as.numeric(crs$rf$predicted))

Data: as.numeric(crs$rf$predicted) in 8768 controls (crs$rf$y FALSE) < 1786 cases (crs$rf$y TRUE).
```

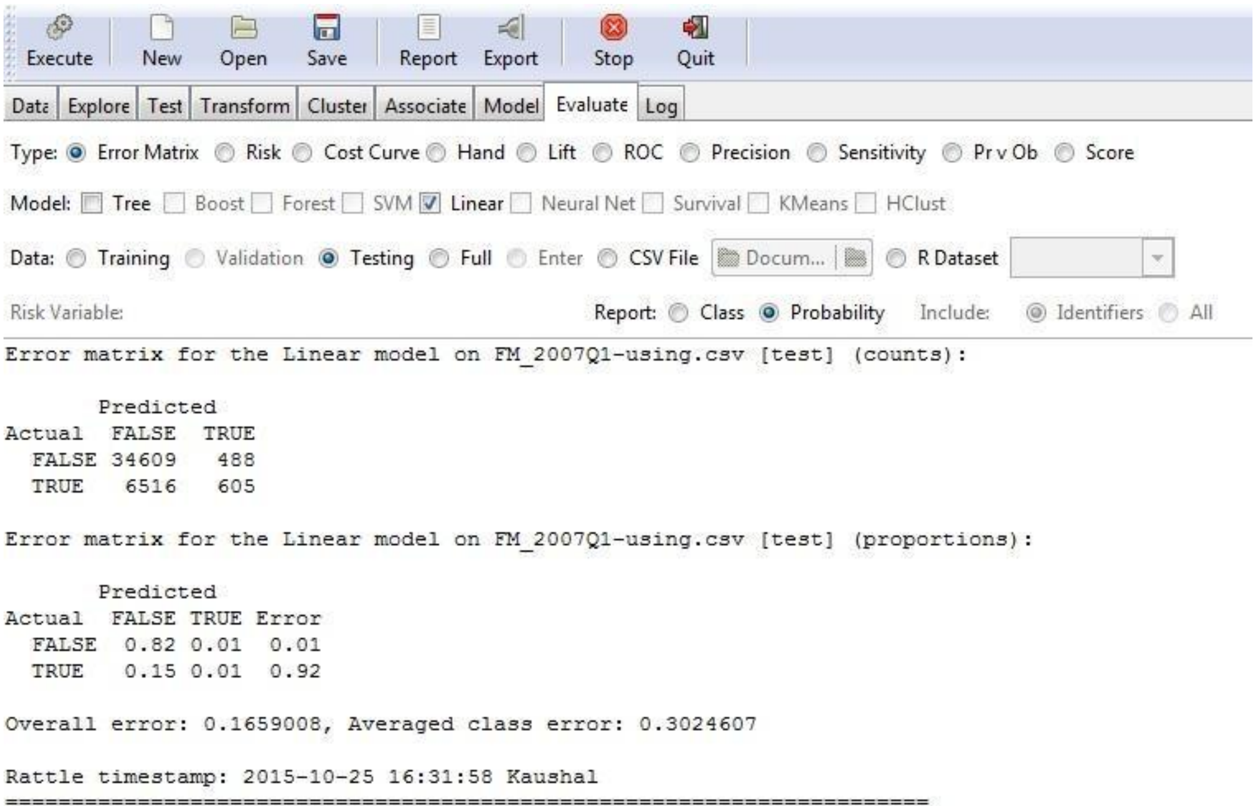
d.) Boost Model




```
Summary of the Ada Boost model:

Call:
ada(DELINQUENCY_STATUS ~ ., data = crs$dataset[crs$train, c(crs$input,
    crs$target)], control = rpart::rpart.control(maxdepth = 30,
    cp = 0.01, minsplit = 20, xval = 10), iter = 50)

Loss: exponential Method: discrete   Iteration: 50

Final Confusion Matrix for Data:
          Final Prediction
True value FALSE TRUE
     FALSE  8628  140
     TRUE   1576  210

Train Error: 0.163

Out-Of-Bag Error:  0.166  iteration= 48

Additional Estimates of number of iterations:

train.err1 train.kap1
        49         50

Variables actually used in tree construction:
 [1] "AZ"                                  "BORROWER_CREDIT_SCORE"
 [3] "CA"                                  "CHANNEL"
 [5] "CLTV"                                "COBORROWER_CREDIT_SCORE"
```

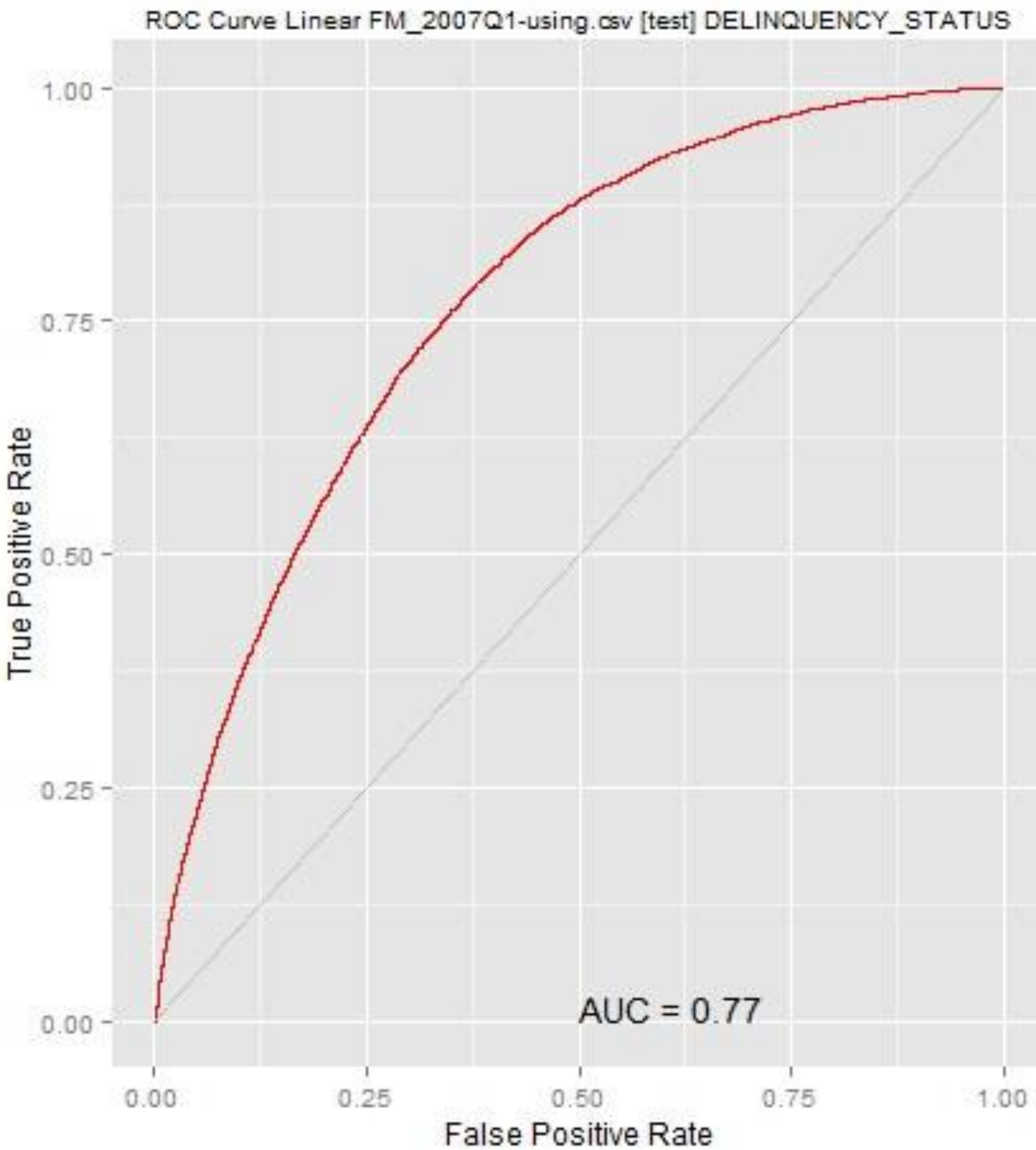
Model performance metrics: FP, FN, Overall error, Sensitivity, Specificity, F1, and AUC. What is the best metric to evaluate model performance?

1.) Error Matrix for Logical Regression Model




```
Error matrix for the Linear model on FM_2007Q1-using.csv [test] (counts):

       Predicted
Actual  FALSE  TRUE
  FALSE 34609   488
  TRUE   6516   605

Error matrix for the Linear model on FM_2007Q1-using.csv [test] (proportions):

       Predicted
Actual  FALSE TRUE Error
  FALSE  0.82 0.01  0.01
  TRUE   0.15 0.01  0.92

Overall error: 0.1659008, Averaged class error: 0.3024607

Rattle timestamp: 2015-10-25 16:31:58 Kaushal
======================================================================
```

ROC Graph for Logical Regression Model



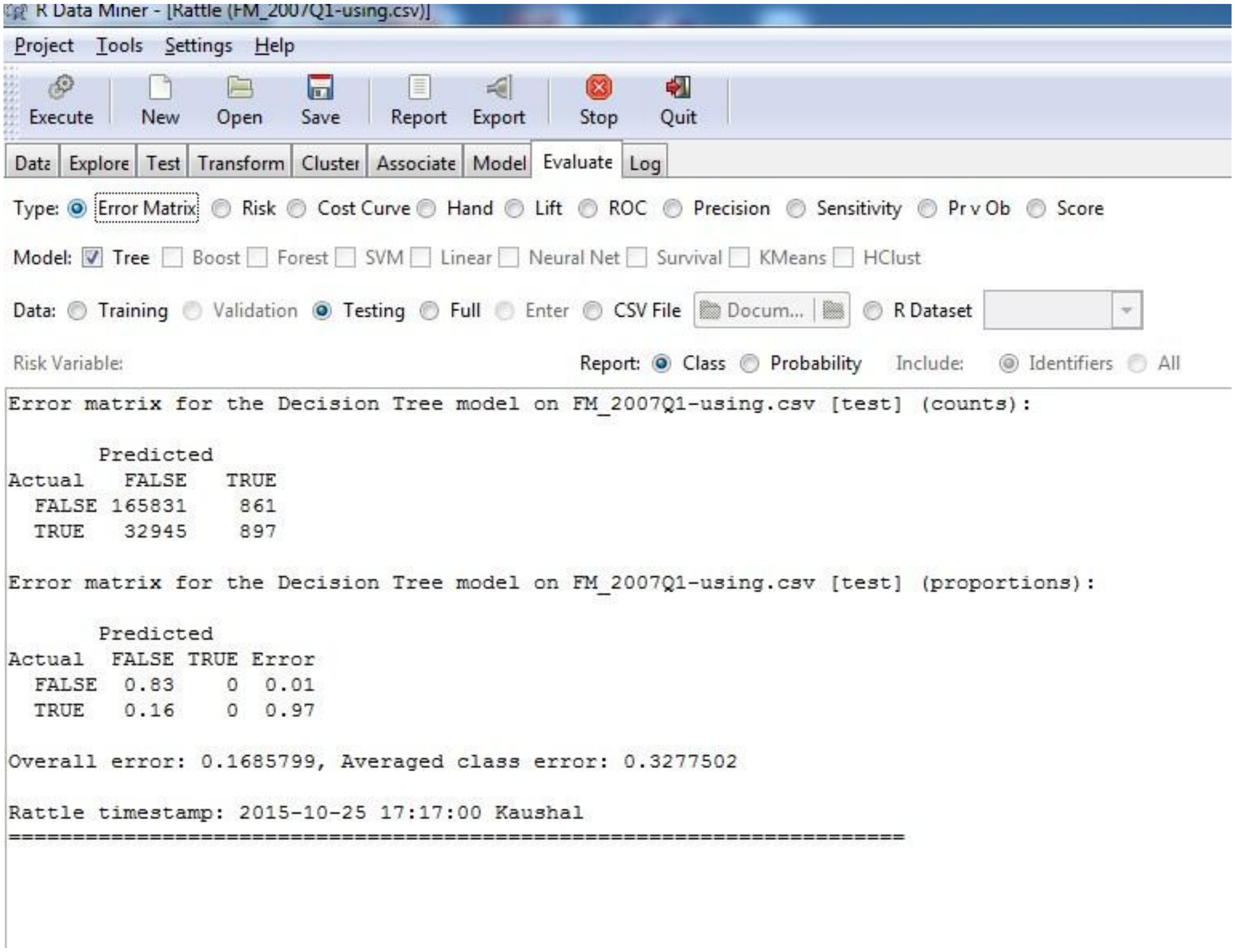
From the Above figures:

False Positive (FP) = 488

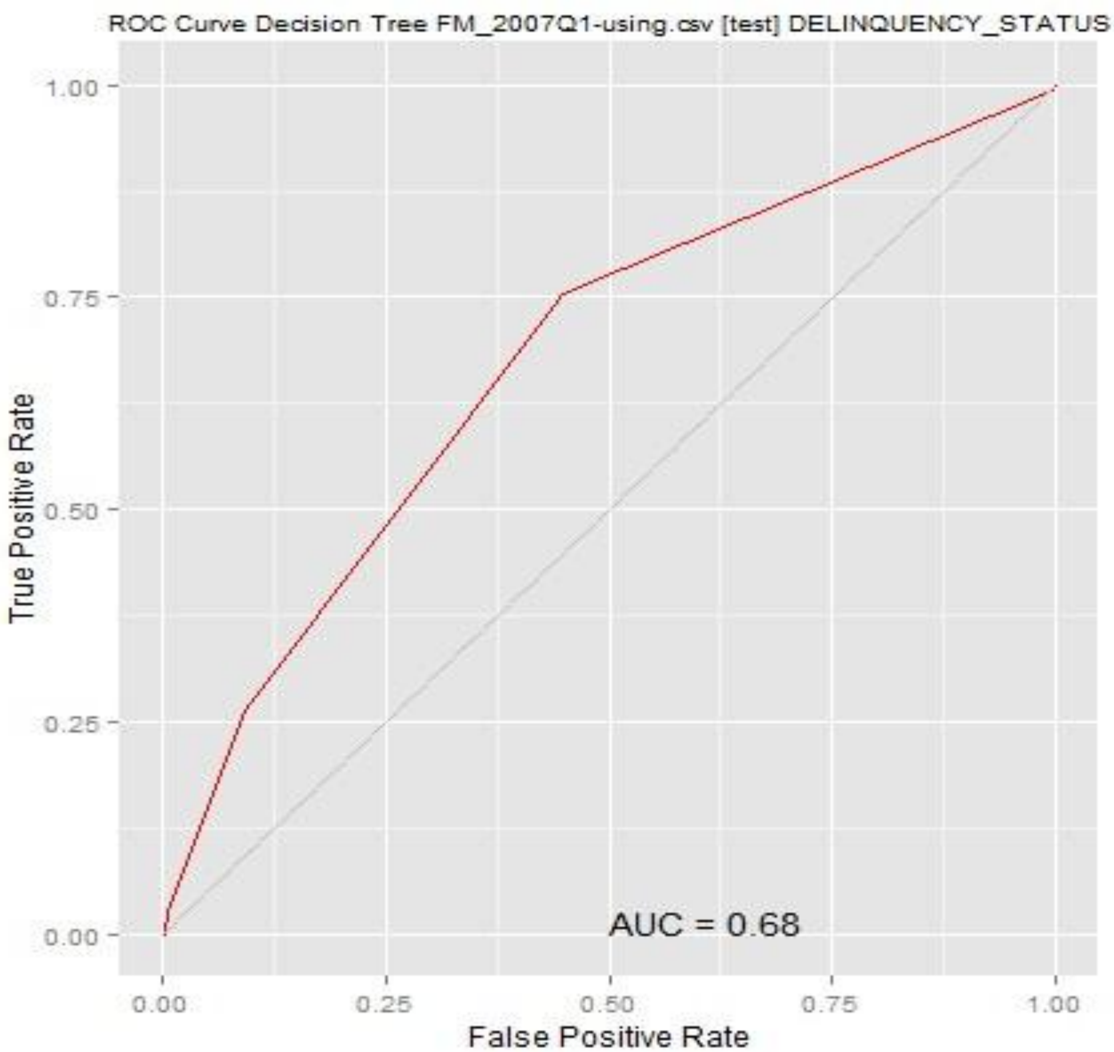
False Negative (FN) = 6516
True Positive (TP) = 605
True Negative (TN) = 34609
Sensitivity = (TP) / ((TP) + (FN)) = 605 / (605 + 6516) = 605/ 7121= 0.08
Specificity = (TN) / ((TN) + (FP)) = 34609 / (34609 + 488) = 34608/35097= 0.98
F1= 2*TP/(2*TP + FP +FN) = 0.147
Overall Error = 0.166

Also from the ROC curve, the **AUC** = 0.77

2.) Error Matrix for Tree Model



ROC Curve of Tree Model



False Positive (FP) = 861
False Negative (FN) = 32945
True Positive (TP) = 897
True Negative (TN) = 165831
Sensitivity = (TP) / ((TP) + (FN)) = 897 / (897 + 32945) = 897/ 33842= 0.03
Specificity = (TN) / ((TN) + (FP)) = 165831 / (165831 + 861) = 34608/35097= 0.99

F1= 2
Overall Error

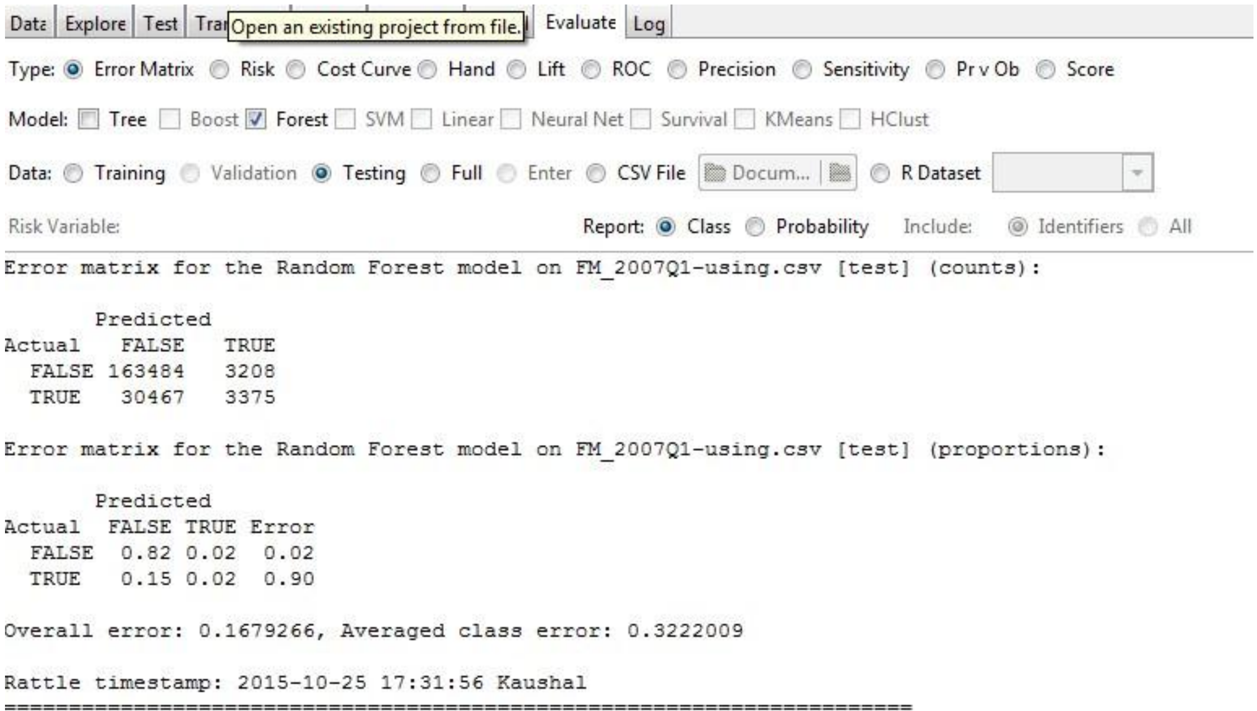
Also fr

*TP/(2*TP + FP +FN) = 0.05 =

0.169 om the ROC curve, the AUC

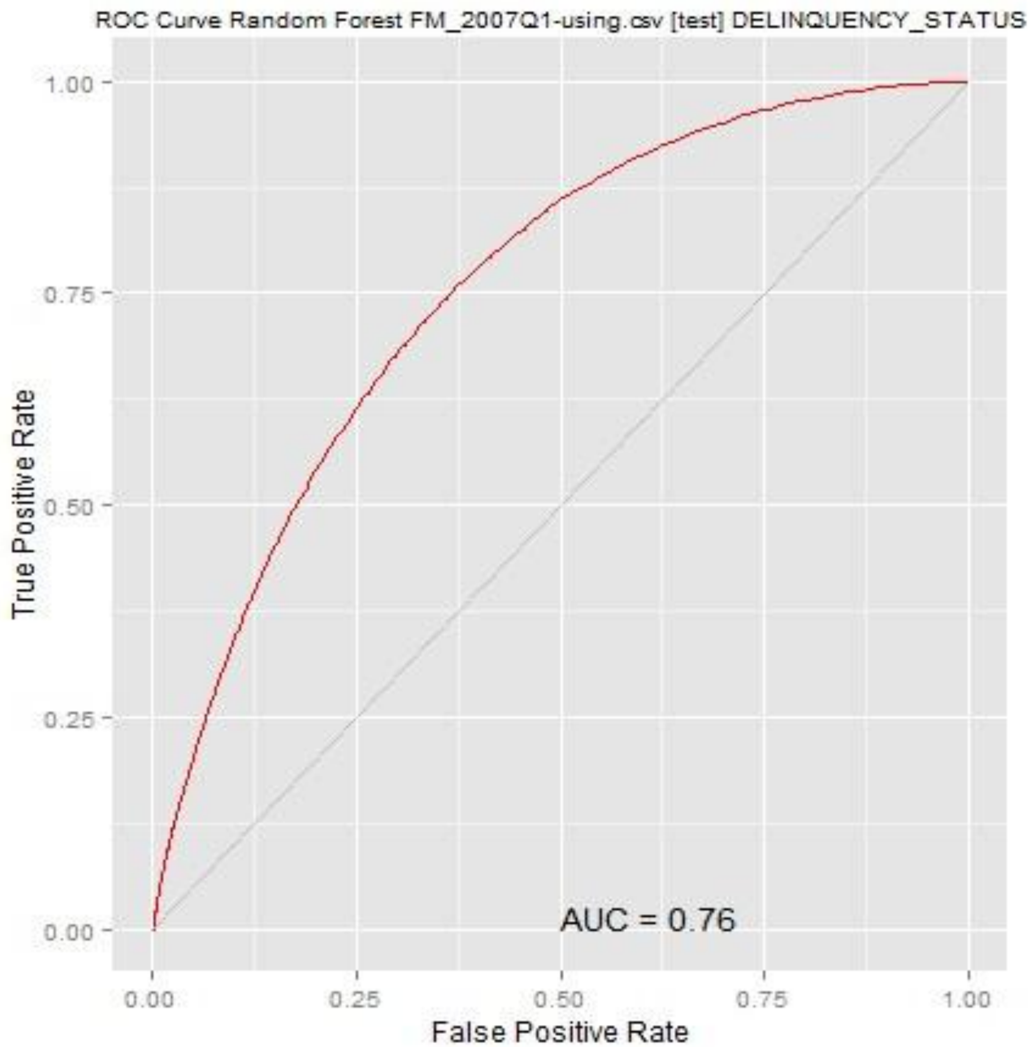
= 0.68

3.) <u>Error Matrix for Forest Tree Model</u>




```
Error matrix for the Random Forest model on FM_2007Q1-using.csv [test] (counts):

       Predicted
Actual   FALSE   TRUE
  FALSE 163484   3208
  TRUE   30467   3375

Error matrix for the Random Forest model on FM_2007Q1-using.csv [test] (proportions):

       Predicted
Actual  FALSE TRUE Error
  FALSE  0.82 0.02  0.02
  TRUE   0.15 0.02  0.90

Overall error: 0.1679266, Averaged class error: 0.3222009

Rattle timestamp: 2015-10-25 17:31:56 Kaushal
======================================================================
```

ROC Curve of Forest Tree Model



From the Above figures:

False Positive (FP) = 3208

False Negative (FN) = 30467

True Positive (TP) = 3375

F1= 2

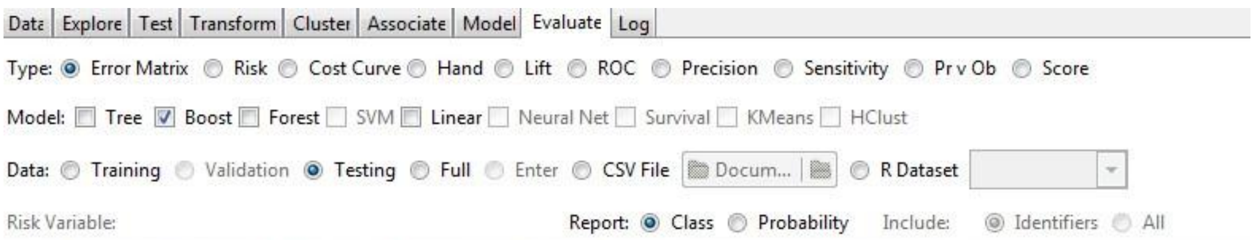
Overall Error

Also fr

True Negative (TN) = 163484

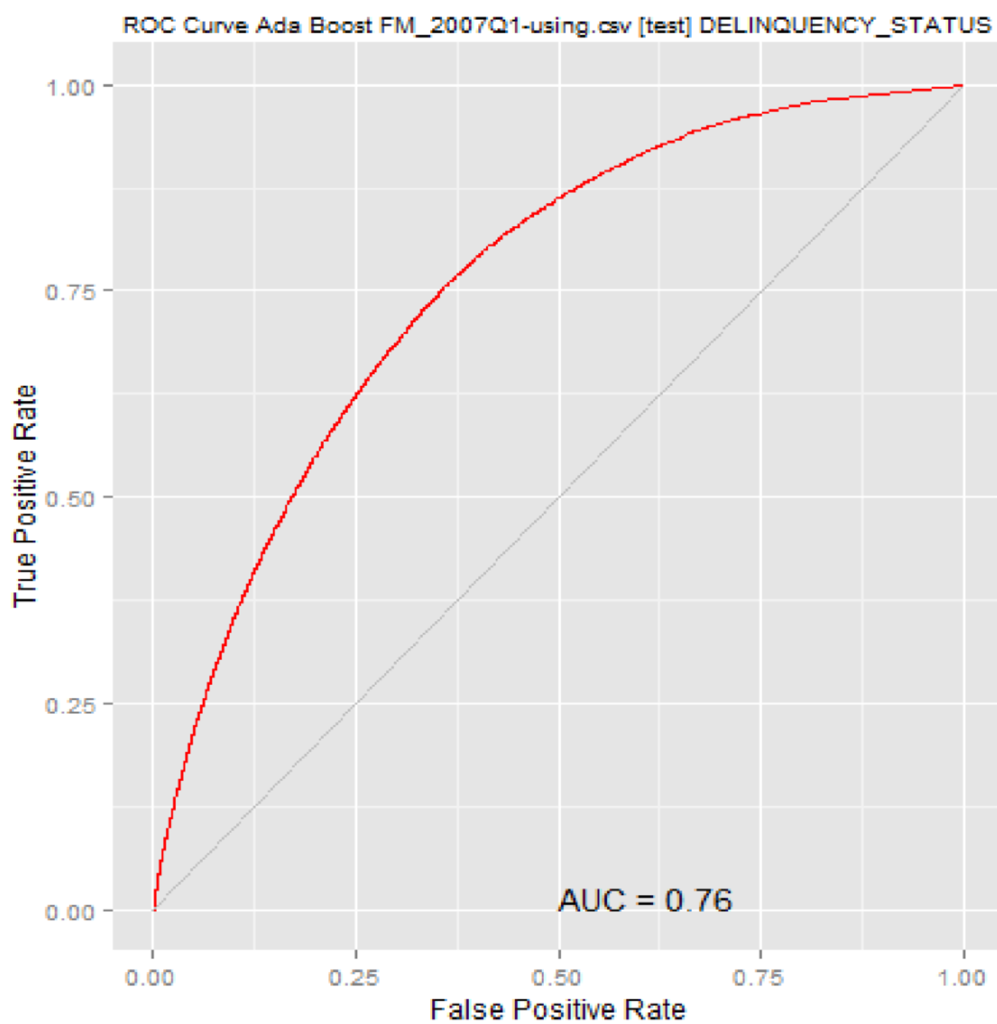
Sensitivity= (TP) / ((TP) + (FN)) = 3375 / (3375 + 30467) = 3375/33842= 0.10
Specificity = (TN) / ((TN) + (FP)) = 163484 / (163484 + 3208) = 0.98
*TP/(2*TP + FP +FN) =0.166 = 0.168 om the ROC curve, the **AUC** = 0.76

4.) <u>Error Matrix for Boost Model</u>




```
Error matrix for the Ada Boost model on FM_2007Q1-using.csv [test] (counts):

       Predicted
Actual   FALSE   TRUE
  FALSE 164025   2667
  TRUE   30812   3030

Error matrix for the Ada Boost model on FM_2007Q1-using.csv [test] (proportions):

       Predicted
Actual  FALSE TRUE Error
  FALSE  0.82 0.01  0.02
  TRUE   0.15 0.02  0.91

Overall error: 0.1669492, Averaged class error: 0.3131418

Rattle timestamp: 2015-10-25 17:41:10 Kaushal
======================================================================
```

ROC Curve of Boost Model



From the Above figures:
False Positive (FP) = 2667

F1= 2
Overall Error

Also fr

False Negative (FN) = 30812
True Positive (TP) = 3030
True Negative (TN) = 164025

Sensitivity = (TP) / ((TP) + (FN)) = 3030 / (3030 + 30812) = 0.089
Specificity = (TN) / ((TN) + (FP)) = 164025/ (164025 + 2667) = 0.98
*TP/(2*TP + FP +FN) = 0.15 =

0.167 om the ROC curve, the **AUC**

= 0.76 **Ans: Sensitivity is the**

best metric for evaluating model

performance.

One of the important criteria that one should consider is the number of false negatives per. Because, our aim is to reduce our losses, by not giving out loans to the parties who will be delinquent in paying back the loan, as the amounts are in several thousands of dollar per candidate and even a single default would result in complete loss of huge sum of money. Also, false positives would only result in losses wrt interests received over the given loan, which are relatively smaller than the losses incurred from delinquency. Thus, we can focus on this issue by developing a model which would result in the least value of false negatives per positive condition i.e. (FN/ (TP + FN)) should be as minimum as possible. Observing the equation we realize that it is nothing but (1-sensitivity). Thus we should consider the model which minimizes (1-sensitivity) or in other words provides highest sensitivity.

Observing the various models, we notice that Forest model provides the best value for the sensitivity.

AUC is another determinant of good model as it plots the graph of sensitivity against specificity, providing a good metric in a graphical format. Thus more the AUC, better would be the model, as it is proportional to sensitivity.

F1= 2
Overall Error

Also fr

3. Did Fannie Mae have information that could have accurately predicted defaults among mortgages issued in Q1 2007?

Ans: On observing the significant variables in the logical regression graph, it is observed that States Arizona, California and Nevada are the prime defaulter states. While doing a brief research of the economic scenario, it comes to our knowledge that during 2007, Arizona and California offered mortgage loans at a very low price of 4% to buy land, owing to the increase in the price of real estate before its ultimate fall. As a result we could predict the high number of applicants in these states, investing in real estate owing to the booming land rates and low loan interest rates. With more people applying for loans, there was increase in demand for real estate, resulting in further rise in housing prices in the market, creating an optimistic scenario. We can see that, the only way this loan could be repaid was by relying on this optimistic trend in real estate, so Fannie Mae could have easily predicted that once this trend stops or reverses, it would result in the rise in defaulters.

Nevada saw huge investments from 2005 and 2007, also it experienced a huge tourist revenue. As a result of this the real estate prices in Nevada soared, with 7% interest rate, there was huge demand in loan application in Nevada along with the resulting increase in demand for real estate resulting in similar scenario as of Arizona and California, that should have been predicted by Fanny Mae.

Observing the tree model, we observe that the default rate increases as the credit score of the defaulter decreases, with almost 32% default rate for people having score below 670. As credit score is a number generated through algorithms by churning huge data, it seems a reliable tool to follow. Thus, Fannie Mae had tools to predict the defaulters and still they provided loans.