Group 5 A03

04/25/2023

Data Mining

Project Report

Predicting Term Deposit Subscriptions via Telemarketing

Problem Description:

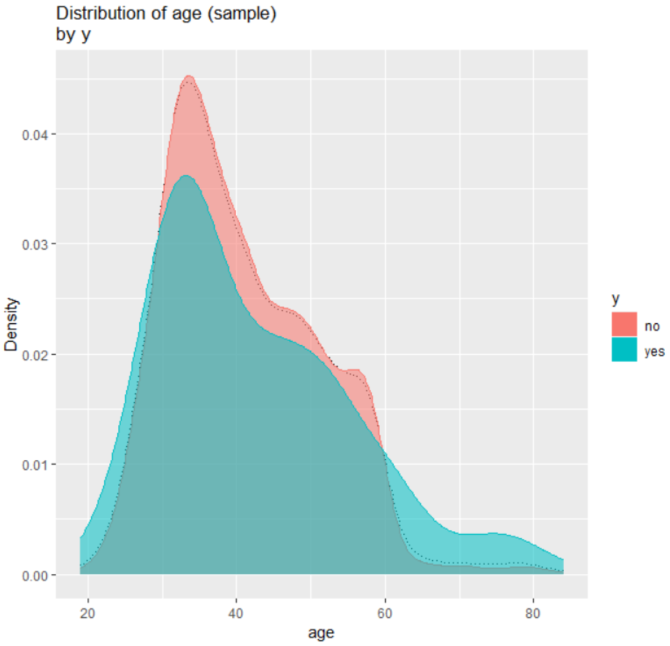
The data we used is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls with clients. We found this data from UCI's Machine Learning Repository. The goal is to predict whether or not the client will subscribe (yes/no) a term deposit (variable y). A term deposit is an investment with a fixed term that requires a deposit of money to the bank. This analysis is meant to improve telemarketing methods in order to increase term deposits at banks.

Data Description:

We did not have to do any data preprocessing as the data came clean and absent of any NA's. As for transformations, the models we utilized did not require rescaling or recoding the data.

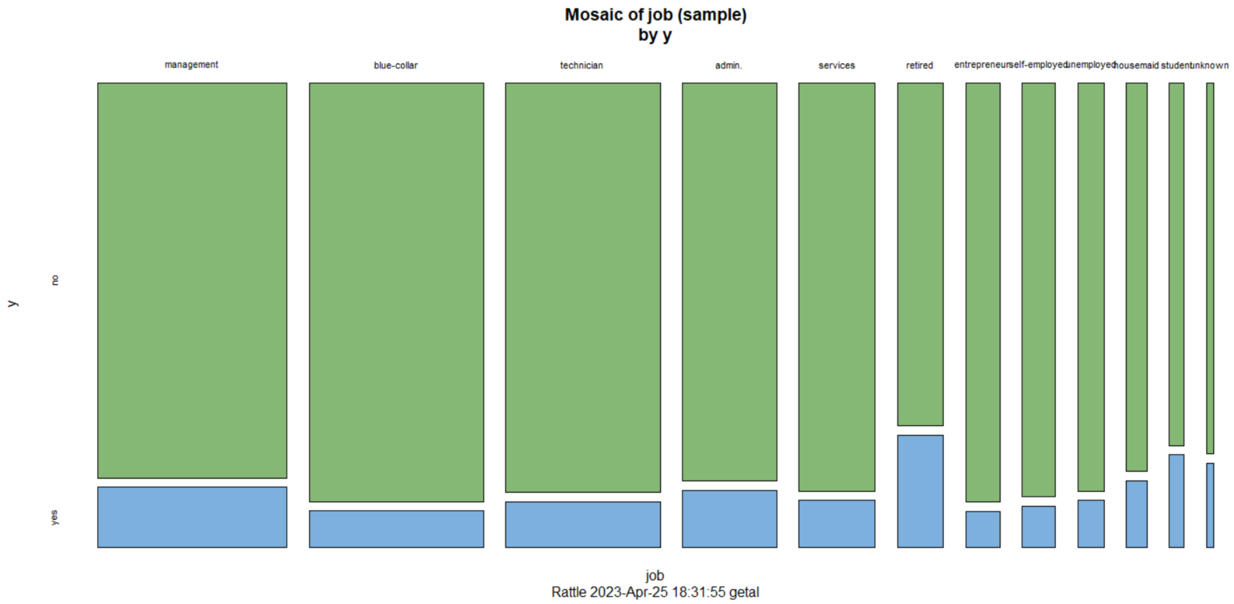
The following input variables represent the banks' client data:

Age: age of the client (numeric)



It appears that customers over 60 years of age have all subscribed to a term loan.

job: type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')


Mosaic of job (sample)
by y

Rattle 2023-Apr-25 18:31:55 getal

This feature has significant variance between job types.

balance: account balance (numeric)

Clients with higher account balances more often subscribe to a term deposit.

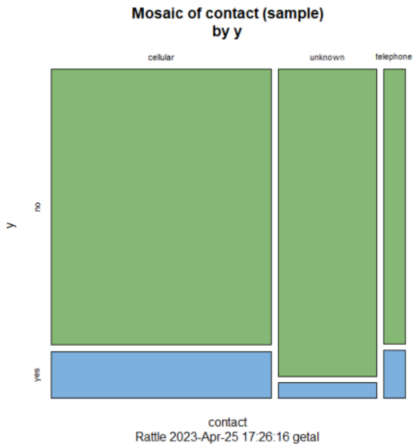
housing: has housing loan? (Categorical: 'no', 'yes', 'unknown')

Those with no housing loans say yes to term deposits more than those who do.

loan: has personal loan? (Categorical: 'no', 'yes', 'unknown')

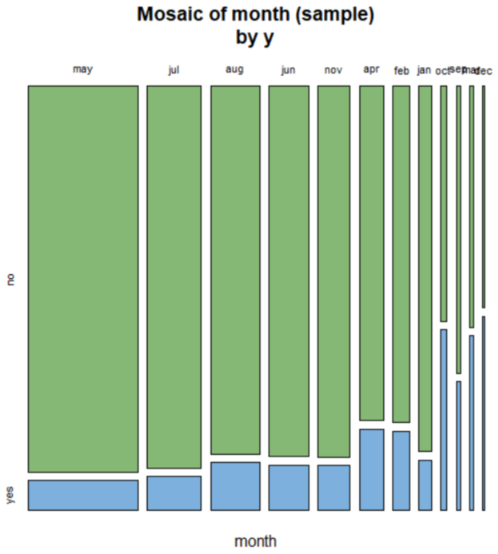
Those with no loans say yes to term deposits more than those who do.

contact: contact communication type (categorical: 'cellular,' 'telephone')



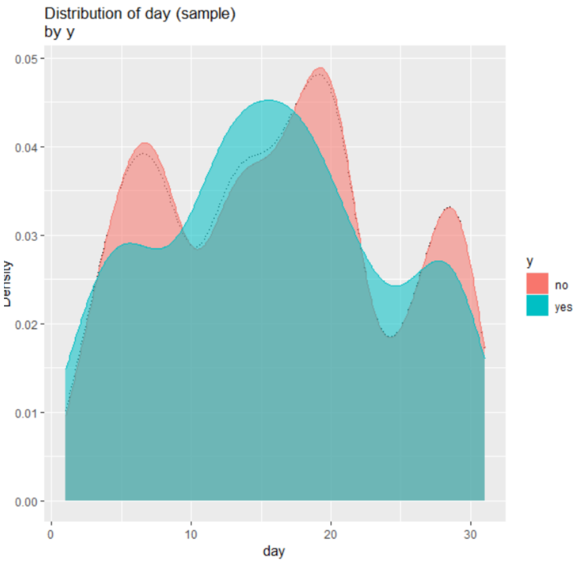
The ‘unknown’ category has significantly less clients who say yes to term deposits.

month: last contact month of year (categorical: 'Jan', 'Feb', 'Mar', ..., 'Nov', 'dec')



Depending on the month the client was last contacted, the number of clients who say yes to a term deposit changes.

day: last contact day of the month (numeric)



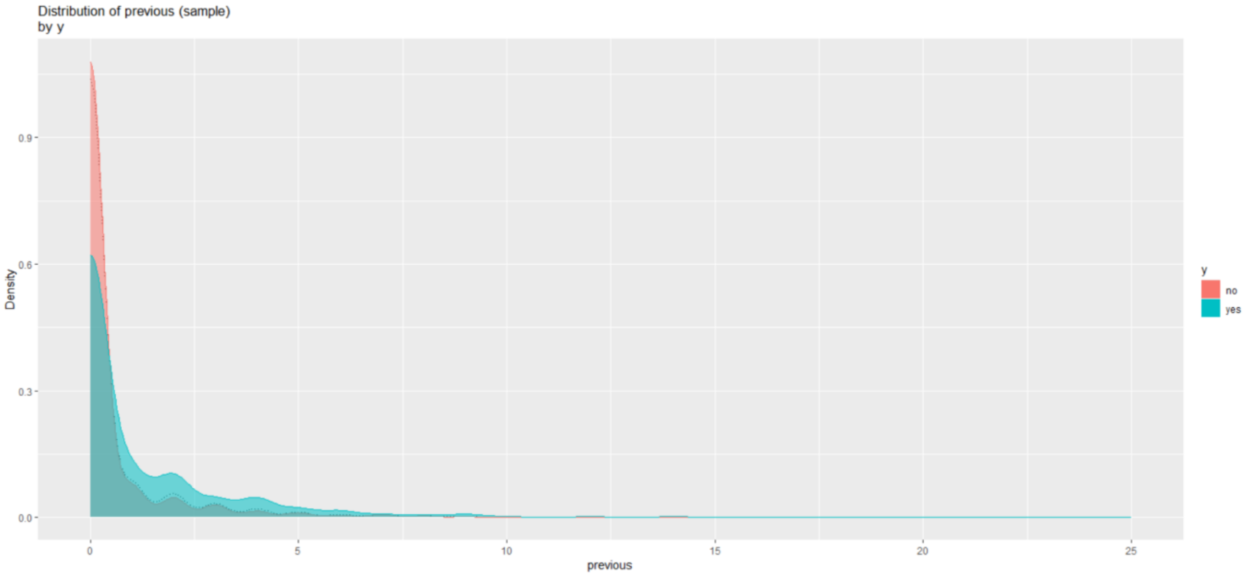
It appears that calls made in the middle of the month are more successful.

campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)

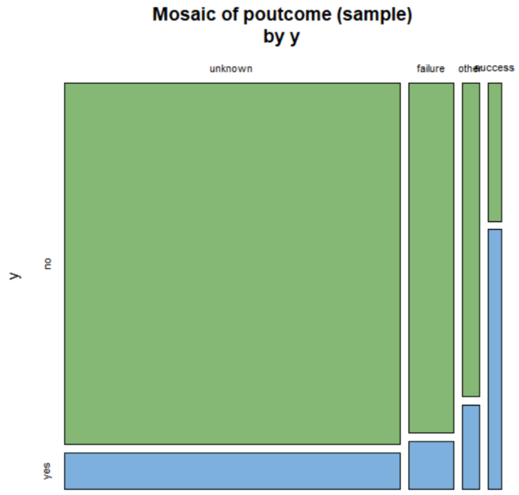
Calls were more successful after waiting a longer period of time.

previous: number of contacts performed before this campaign and for this client (numeric)



The more contacts performed result in more successful calls

poutcome: outcome of the previous marketing campaign (categorical: 'failure', 'nonexistent', 'success')



If the previous marketing campaign was successful, a higher proportion of clients say yes to a term deposit.

Removed Variables

The following categoric variables did not show a significant difference in the output variable between categories:

marital: marital status (categorical: 'divorced', 'married', 'single,' 'unknown'; note: 'divorced' means divorced or widowed)

education (categorical: 'secondary,' 'tertiary,' 'primary,' 'unknown')

default: has credit in default? (Categorical: 'no', 'yes', 'unknown')

duration: last contact duration, in seconds (numeric). This feature was removed because it is not known before a call is performed.

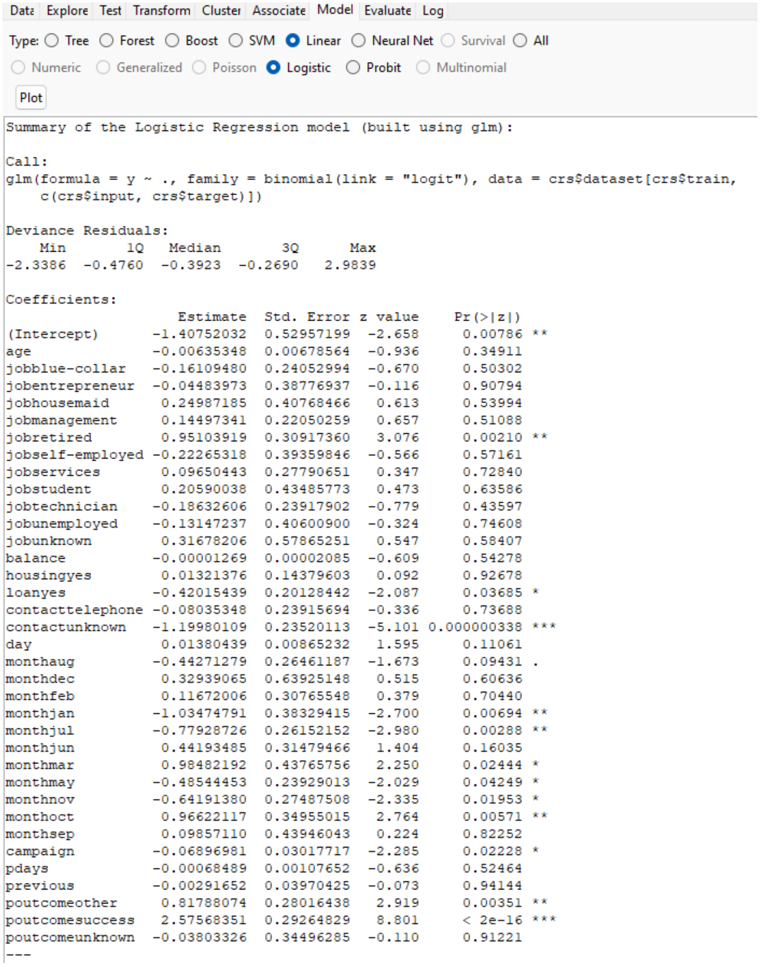
Output variable (desired target):

y - Has the client subscribed to a term deposit? (Binary: 'yes', 'no')

Modeling Description:

We experimented with multiple different models on our data. We ran booth boosts, logistic regression, forest, and neural network. After evaluating all models, we determined that the adaptive boost and logistic regression models worked best for our data, gave the best results, and had the least amount of issues with parameter settings and overfitting. The extreme boost and neural network models consistently returned models that were either overfitting or the discrepancy between the training and validation numbers were large. Additionally, the forest model didn't have as good of an AUC compared to the logistic regression and adaptive boost models. Therefore, we decided to move forward with and evaluate the logistic regression and adaptive boost models.

First, we ran a logistic regression model and the results of it can be seen below. No parameter tuning was needed because of the model chosen. The most important parts of the output include the coefficients (Estimate) and p-values (Pr(>|x|)) for all the variables that we included in the final model.

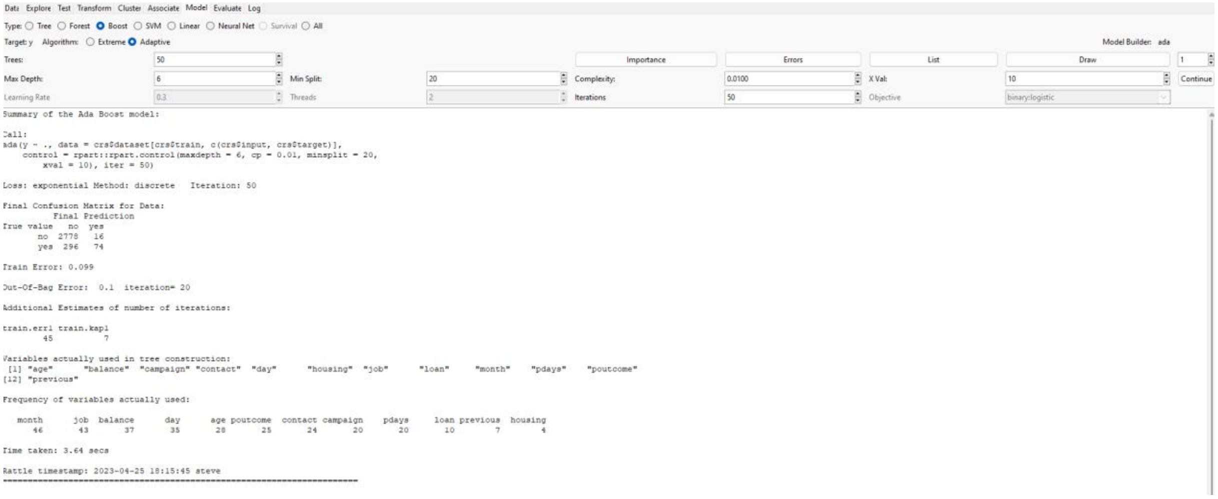

Data Explore Test Transform Cluster Associate Model Evaluate Log
Plot

Type: ○ Tree ○ Forest ○ Boost ○ SVM ● Linear ○ Neural Net ○ Survival ○ All

○ Numeric ○ Generalized ○ Poisson ● Logistic ○ Probit ○ Multinomial

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

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

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3386  -0.4760  -0.3923  -0.2690   2.9839

Coefficients:
                    Estimate  Std. Error z value    Pr(>|z|)
(Intercept)      -1.40752032  0.52957199  -2.658     0.00786 **
age              -0.00635348  0.00678564  -0.936     0.34911
jobblue-collar   -0.16109480  0.24052994  -0.670     0.50302
jobentrepreneur  -0.04483973  0.38776937  -0.116     0.90794
jobhousemaid      0.24987185  0.40768466   0.613     0.53994
jobmanagement     0.14497341  0.22050259   0.657     0.51088
jobretired        0.95103919  0.30917360   3.076     0.00210 **
jobself-employed -0.22265318  0.39359846  -0.566     0.57161
jobservices       0.09650443  0.27790651   0.347     0.72840
jobstudent        0.20590038  0.43485773   0.473     0.63586
jobtechnician    -0.18632606  0.23917902  -0.779     0.43597
jobunemployed    -0.13147237  0.40600900  -0.324     0.74608
jobunknown        0.31678206  0.57865251   0.547     0.58407
balance          -0.00001269  0.00002085  -0.609     0.54278
housingyes        0.01321376  0.14379603   0.092     0.92678
loanyes          -0.42015439  0.20128442  -2.087     0.03685 *
contacttelephone -0.08035348  0.23915694  -0.336     0.73688
contactunknown   -1.19980109  0.23520113  -5.101 0.000000338 ***
day               0.01380439  0.00865232   1.595     0.11061
monthaug         -0.44271279  0.26461187  -1.673     0.09431 .
monthdec          0.32939065  0.63925148   0.515     0.60636
monthfeb          0.11672006  0.30765548   0.379     0.70440
monthjan         -1.03474791  0.38329415  -2.700     0.00694 **
monthjul         -0.77928726  0.26152152  -2.980     0.00288 **
monthjun          0.44193485  0.31479466   1.404     0.16035
monthmar          0.98482192  0.43765756   2.250     0.02444 *
monthmay         -0.48544453  0.23929013  -2.029     0.04249 *
monthnov         -0.64191380  0.27487508  -2.335     0.01953 *
monthoct          0.96622117  0.34955015   2.764     0.00571 **
monthsep          0.09857110  0.43946043   0.224     0.82252
campaign         -0.06896981  0.03017717  -2.285     0.02228 *
pdays            -0.00068489  0.00107652  -0.636     0.52464
previous         -0.00291652  0.03970425  -0.073     0.94144
poutcomeother     0.81788074  0.28016438   2.919     0.00351 **
poutcomesuccess   2.57568351  0.29264829   8.801     < 2e-16 ***
poutcomeunknown  -0.03803326  0.34496285  -0.110     0.91221
---
```

Secondly, we ran an adaptive boost model and the results of that can be seen below as well. We experimented with changing the max depth and the number of trees to see if that helped the model and evaluation of the model at all. However, none of the changes made much of a significant difference in the evaluation on training or validation, so we decided the default parameter settings were the best option for the model.



Type: ○ Tree ○ Forest ● Boost ○ SVM ○ Linear ○ Neural Net ○ Survival ○ All

Target: y Algorithm: ○ Extreme ● Adaptive

Model Builder: ada

Trees: 50

Max Depth: 6 Min Split: 20 Complexity: 0.0100 X Val: 10

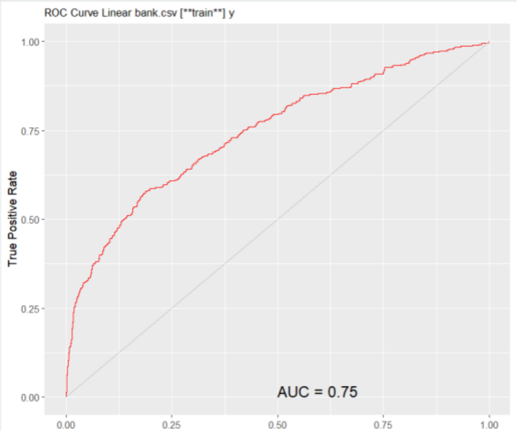
Learning Rate 0.3 Threads 2 Iterations 50 Objective binary:logistic

Draw: 1

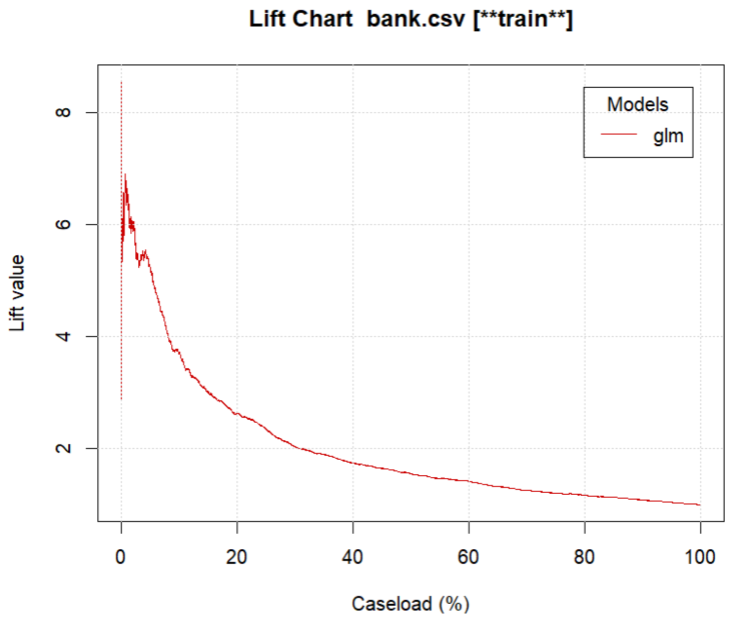
```
Summary of the Ada Boost model:

Call:
ada(y ~ ., data = crs$dataset[crs$train, c(crs$input, crs$target)],
    control = rpart::rpart.control(maxdepth = 6, cp = 0.01, minsplit = 20,
        xval = 10), iter = 50)

Loss: exponential Method: discrete   Iteration: 50

Final Confusion Matrix for Data:
          Final Prediction
True value   no  yes
       no  2778   16
       yes  296   74

Train Error: 0.099

Out-Of-Bag Error:  0.1  iteration= 20

Additional Estimates of number of iterations:

train.err1 train.kap1
        45          7

Variables actually used in tree construction:
 [1] "age"      "balance"  "campaign" "contact"  "day"      "housing"  "job"      "loan"     "month"    "pdays"    "poutcome"
[12] "previous"

Frequency of variables actually used:

   month      job  balance      day      age poutcome  contact campaign    pdays     loan previous  housing
      46       43       37       35       28       25       24       20       20       10        7        4

Time taken: 3.64 secs

Rattle timestamp: 2023-04-25 18:15:45 steve
======================================================================
```

Evaluation:

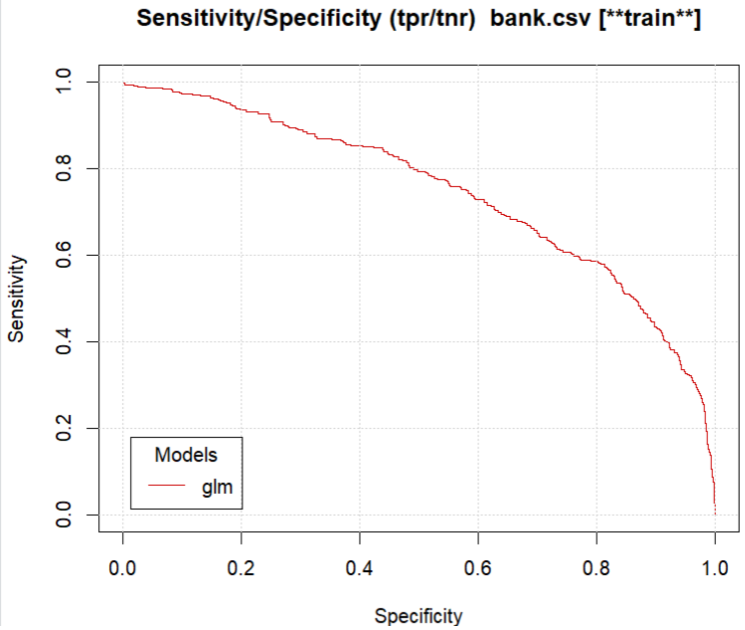
For this data set, we used multiple tests to run evaluations on the target variable. First, we ran ROC, Lift, and Sensitivity evaluations on the training set using **Logistic Regression**. Using the logistic regression to determine the ROC which ran as a score of 0.7460.



Next, we ran the Lift score which measures the effectiveness of models by calculating the ratio between the result obtained with a model and the result obtained without a model and also identifies the gain in performance offered by the model.

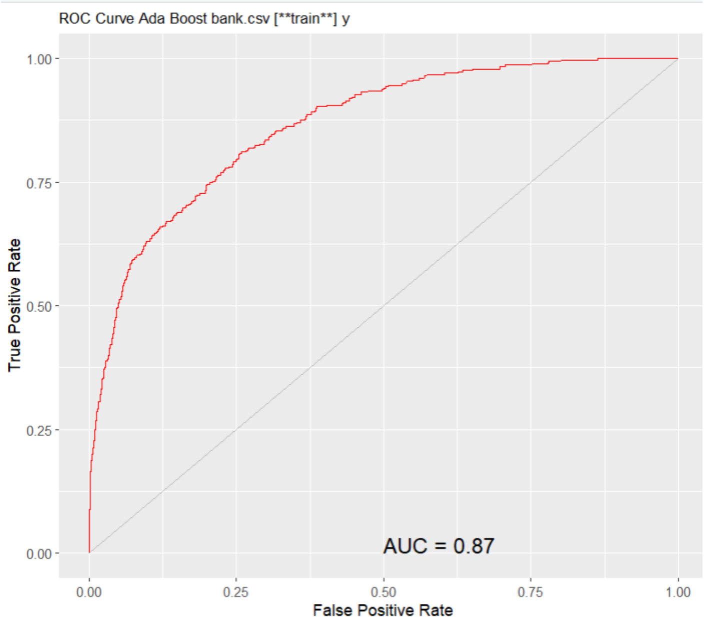


Finally, with the logistic regression, we ran a Sensitivity evaluation on the training data and found the following:

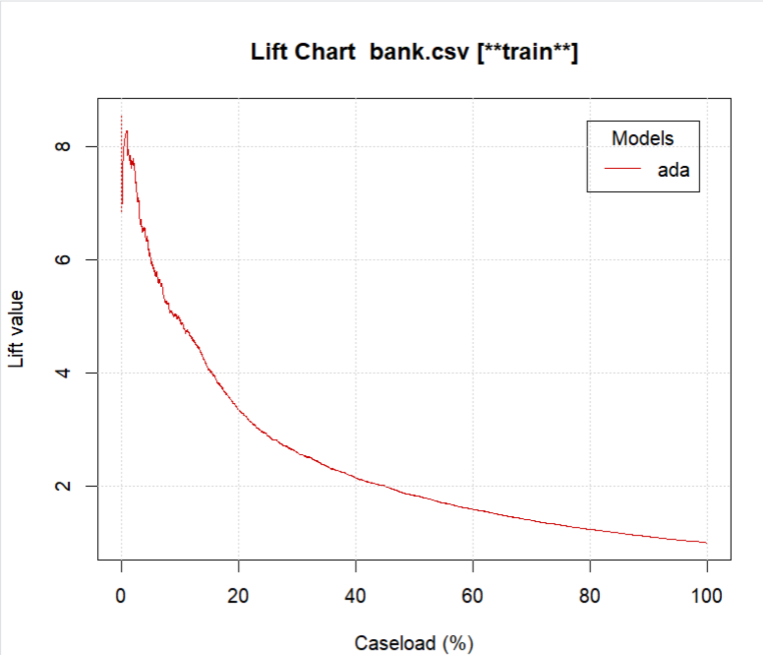


A sensitivity/specificity chart ranks the assumptions from the most important down to the least important in the model. If an assumption and a forecast have a high correlation coefficient, it means that the assumption has a significant impact on the forecast.

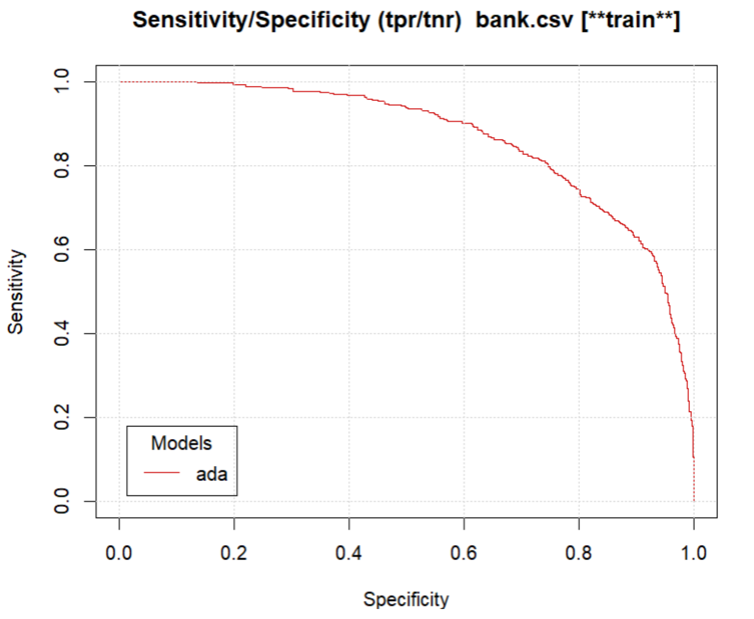
After running those three evaluations on the logistic regression of the training data, we evaluated the **Adaptive Boost Model** on the training data. Getting the ROC score of this model equates to 0.8651 which is higher than the linear logistic ROC score.



The lift chart of the boosted model shows a slightly higher curve than the logistic regression as shown by the dip in the beginning of the chart starting higher than the logistic regression.



Finally, we did the sensitivity chart of the boosted model shows more of a smooth curve downwards compared to the rushed downward curve of the logistic regression's sensitivity chart.



The best model was identified as the ROC of the adaptive boosted model as it is 0.12 points higher than the logistic regression ROC score. We found this by running multiple scores and finding the one that fit the data the best. Also, feature selection was important in finding the best ROC. We ran multiple models and iterations and decided that AdaBoost was the best. Overall, we are happy with the ROC score that the boosted model provides as it is a good indicator for the target variable.

Recommendations:

We learned that building a relationship with a client, or contacting clients consistently, is an important factor in whether or not that client will subscribe to a term deposit. In addition to that, job type and age are also important factors. Retired people over 60 years of age are more likely to say yes to a term deposit. This model should be used to make marketing campaigns more effective by targeting older, retired clients. Those clients should also be contacted consistently, but not too often. Moving forward, I would suggest collecting more information on clients to add more features to the model. Also, you could utilize different models.

Dataset Source:

https://archive.ics.uci.edu/ml/datasets/bank+marketing

[Moro et al., 2014] S. Moro, P. Cortez, and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014