Final Project

Andrew Barlow

Introduction

A Portuguese bank collected data from 2008 - 2013 on a direct telemarketing campaign to get customers to subscribe to a bank term deposit. Due to the 2008 financial crisis, European banks were pressured to increase capital requirements; this event led the firm to seek more long term depositors. Out of 30,488 calls only 3,859 customers subscribed to a term deposit, which is a success rate of 12.65%. This is not optimal for the company as too much time and resources of the call centers was spent on an unsuccessful telemarketing campaign. The goal of this research project is to determine if a decision support system based on machine learning models can be implemented to optimize the telemarketing team's time and resource allocation. A robust exploration of data mining techniques were implemented to determine an optimal model to be used in the decision support system.

Statement of Business Problem

The Portuguese bank needs to improve their telemarketing caimpaign in order to gain more long term depositors. The main focus of this paper is to determine if a machine learning model can accurately predict if an individual will accept or deny a term deposit subscription. If a machine learning model can accurately classify an individual who will accept a term deposit subscription, then it could be used on new data to better target individuals who will have a higher probability of accepting a deposit subscription with the bank. This machine learning model could theoretically be used in a decision support system to help telemarketers target individuals with a higher probability of accepting a term deposit subscription. The paper will focuse on logistic regression, linear discriminant analysis, and naive Bayes as well as modeling rare events.

Data Sources & Data Description

This dataset comes from the UCI machine learning repository, which was donated in 2014 from a Portuguese bank. Here is a link to the data: https://archive.ics.uci.edu/ml/datasets/Bank+Marketing. I am using a subset of this dataset that includes 30,488 observations. The dataset includes 20 features and one target feature named Subscribed. The features include socioeconomic variables such as age, education, job, personal loan, housing loan, etc. It also includes features pertaining to the calls such as communication type, month, day of week, duration, etc. A major caveat of this dataset is the feature called 'duration'. This feature greatly affects the target of Subscribed, because if duration = 0, then Subscribed is obviously 'no'. Also, the duration of the call is not known before a call is performed. All of the other features in the dataset are known before the call, except for duration and Subscribed. At the end of each call Subscribed is obviously known, so we will drop duration when completing our analysis. Next I will perform some EDA to get a better sense of the features and the class balance of our target.

Exploratory Data Analysis

Create BANK Data Frame

```
PATH="C:/Users/Andrew/Documents/R"
BANK<- read.csv(file.path(PATH,"Bank.csv"), header=TRUE)
attach(BANK)
```

Pre-Processing Tasks

```
table(Subscribed)
```

```
## Subscribed
##    no   yes
## 26629  3859
```

We observe that the way our target variable is coded is not what we want because R is assuming that "no" is our class of interest. As discussed in lecture, this will flip our results, meaning sensitivity and specificity measures for predicting "yes" and "no" will be flipped. To fix this, we will re-code our target variable Subscribed. The new levels of the factor variable will be "accept" and "decline", where accept represents a customer who accepted a term deposit subscription and "decline" represents a customer who declined a term deposit subscription.

```
# Re-code binary target with new levels
BANK$Subscribed<- as.factor(ifelse(BANK$Subscribed=="no", "decline", "accept"))
# Hard code our binary target
SUBSCRIBED <- ifelse (BANK$Subscribed=="decline", 0,1)
# Combine columns
BANK<-cbind(BANK,SUBSCRIBED)
# Drop duration
BANK<- BANK[c(-11)]
# Rename Subscribed as true_class
names(BANK)[20]<-"true_class"
# Re-attach dataframe
attach(BANK)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     SUBSCRIBED
```

```
## The following objects are masked from BANK (pos = 3):
##
##     age, campaign, cons.conf.idx, cons.price.idx, contact,
##     day_of_week, default, education, emp.var.rate, euribor3m,
##     housing, job, loan, marital, month, nr.employed, pdays,
##     poutcome, previous
```

Generate Descriptive Statistics

```
library(psych)
describe(BANK)
```

```
##                vars     n    mean     sd  median trimmed   mad     min
## age               1 30488   39.03  10.33   37.00   38.12  8.90   17.00
## job*              2 30488    4.72   3.61    3.00    4.49  2.97    1.00
## marital*          3 30488    2.19   0.62    2.00    2.24  0.00    1.00
## education*        4 30488    4.84   1.99    4.00    5.03  2.97    1.00
## default*          5 30488    1.00   0.01    1.00    1.00  0.00    1.00
## housing*          6 30488    1.54   0.50    2.00    1.55  0.00    1.00
## loan*             7 30488    1.16   0.36    1.00    1.07  0.00    1.00
## contact*          8 30488    1.33   0.47    1.00    1.29  0.00    1.00
## month*            9 30488    5.25   2.38    5.00    5.33  2.97    1.00
## day_of_week*     10 30488    3.02   1.40    3.00    3.02  1.48    1.00
## campaign         11 30488    2.52   2.72    2.00    1.96  1.48    1.00
```

```
## pdays            12 30488  956.33 201.37  999.00  999.00  0.00    0.00
## previous         13 30488    0.19   0.52    0.00    0.07  0.00    0.00
## poutcome*        14 30488    1.93   0.38    2.00    1.98  0.00    1.00
## emp.var.rate     15 30488   -0.07   1.61    1.10    0.10  0.44   -3.40
## cons.price.idx   16 30488   93.52   0.59   93.44   93.52  0.82   92.20
## cons.conf.idx    17 30488  -40.60   4.79  -41.80  -40.78  6.52  -50.80
## euribor3m        18 30488    3.46   1.78    4.86    3.61  0.16    0.63
## nr.employed      19 30488 5160.81  75.16 5191.00 5172.08 55.00 4963.60
## true_class*      20 30488    1.87   0.33    2.00    1.97  0.00    1.00
## SUBSCRIBED       21 30488    0.13   0.33    0.00    0.03  0.00    0.00
##                    max  range   skew kurtosis   se
## age              95.00  78.00   0.98     1.24 0.06
## job*             11.00  10.00   0.41    -1.47 0.02
## marital*          3.00   2.00  -0.16    -0.57 0.00
## education*        7.00   6.00  -0.36    -1.12 0.01
## default*          2.00   1.00 100.79 10157.00 0.00
## housing*          2.00   1.00  -0.17    -1.97 0.00
## loan*             2.00   1.00   1.89     1.58 0.00
## contact*          2.00   1.00   0.73    -1.47 0.00
## month*           10.00   9.00  -0.30    -1.07 0.01
## day_of_week*      5.00   4.00   0.00    -1.27 0.01
## campaign         43.00  42.00   4.90    38.16 0.02
## pdays           999.00 999.00  -4.51    18.32 1.15
## previous          7.00   7.00   3.59    17.68 0.00
## poutcome*         3.00   2.00  -0.73     3.26 0.00
## emp.var.rate      1.40   4.80  -0.55    -1.27 0.01
## cons.price.idx   94.77   2.57  -0.12    -0.86 0.00
## cons.conf.idx   -26.90  23.90   0.37    -0.32 0.03
## euribor3m         5.04   4.41  -0.52    -1.63 0.01
## nr.employed    5228.10 264.50  -0.89    -0.35 0.43
## true_class*       2.00   1.00  -2.25     3.05 0.00
## SUBSCRIBED        1.00   1.00   2.25     3.05 0.00
```

Examine Class Balance

```
table(true_class)
```

```
## true_class
##  accept decline
##    3859   26629
```

We observe that the outcome feature 'Subscribed' is unbalanced. Out of the 30,488 customers that were called, only 3,859 accepted a term deposit subscription. Roughly 12.66% of customers accepted a deposit subscription and 87.34% declined a deposit subscription. This is the case in most business binary classification problems. Later in the paper we will utilize modeling rare events techniques to compare the predictive accuracies of the models. First we will use the unbalanced dataset, and then use a balanced version.

Examine Other Categorical Variables

```
table(true_class, loan)
```

```
##           loan
## true_class    no   yes
##    accept   3274   585
```

```
##    decline 22446  4183
```

Prob[accept | Yes loan] = [585/4768] = 12.27%

Prob[accept | No loan] = [3274/25720] = 12.73%

Since the probability of accepting a deposit subscription does not seem to be affected by whether or not a customer has a personal loan with the bank, we can reasonably conclude that having a personal loan with the bank-holding all else equal-does not affect the probability of accepting a deposit subscription.

```
table(true_class, housing)
```

```
##           housing
## true_class    no   yes
##    accept   1717  2142
##    decline 12250 14379
```

Prob[accept | Yes housing] = [2142/16521] = 12.96%

Prob[accept | No housing] = [1717/13967] = 12.29%

We can also conclude that having a housing loan with the bank-holding all else equal-does not affect the probability of accepting a deposit subscription.

Statistical Analysis

First we will randomize the data and create our training and testing sets

```
set.seed(123)
RANDOMIZED_DATA <- BANK[order(runif(30488)), ]
# 70% training and 30% testing
TRAINING_SET<- RANDOMIZED_DATA[1:21342, ]
TEST_SET<- RANDOMIZED_DATA[21343:30488, ]
```

Logistic Regression

Logit Model (Full Model)

```
attach(TRAINING_SET)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     SUBSCRIBED
```

```
## The following objects are masked from BANK (pos = 4):
##
##     age, campaign, cons.conf.idx, cons.price.idx, contact,
##     day_of_week, default, education, emp.var.rate, euribor3m,
##     housing, job, loan, marital, month, nr.employed, pdays,
##     poutcome, previous, SUBSCRIBED, true_class
```

```
## The following objects are masked from BANK (pos = 5):
##
##     age, campaign, cons.conf.idx, cons.price.idx, contact,
##     day_of_week, default, education, emp.var.rate, euribor3m,
##     housing, job, loan, marital, month, nr.employed, pdays,
##     poutcome, previous
```

```r
FULL_MODEL <- glm(SUBSCRIBED~. -true_class, family=binomial(), data = TRAINING_SET)

# Implement Stargazer library to view results
library(stargazer)
```

```
##
## Please cite as:
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```r
stargazer(FULL_MODEL, type = "text")
```

```
##
## ===============================================================
##                                    Dependent variable:
##                                -----------------------------
##                                          SUBSCRIBED
## ---------------------------------------------------------------
## age                                        -0.003
##                                            (0.003)
##
## jobblue-collar                             -0.136
##                                            (0.095)
##
## jobentrepreneur                            -0.042
##                                            (0.145)
##
## jobhousemaid                                0.125
##                                            (0.174)
##
## jobmanagement                              -0.104
##                                            (0.100)
##
## jobretired                                 0.299**
##                                            (0.130)
##
## jobself-employed                           -0.023
##                                            (0.130)
##
## jobservices                                -0.083
##                                            (0.100)
##
## jobstudent                                  0.210
##                                            (0.136)
##
## jobtechnician                               0.036
##                                            (0.081)
##
## jobunemployed                              -0.124
##                                            (0.150)
##
## maritalmarried                             -0.039
##                                            (0.080)
```

```
##
## maritalsingle                          -0.004
##                                        (0.090)
##
## educationbasic.6y                       0.160
##                                        (0.150)
##
## educationbasic.9y                       0.028
##                                        (0.116)
##
## educationhigh.school                    0.046
##                                        (0.112)
##
## educationilliterate                     0.297
##                                        (1.175)
##
## educationprofessional.course            0.067
##                                        (0.122)
##
## educationuniversity.degree              0.161
##                                        (0.112)
##
## defaultyes                             -8.468
##                                      (113.372)
##
## housingyes                             -0.043
##                                        (0.048)
##
## loanyes                                 0.038
##                                        (0.065)
##
## contacttelephone                     -0.736***
##                                        (0.086)
##
## monthaug                             0.587***
##                                        (0.141)
##
## monthdec                              0.494**
##                                        (0.239)
##
## monthjul                                0.144
##                                        (0.111)
##
## monthjun                             -0.766***
##                                        (0.144)
##
## monthmar                             1.708***
##                                        (0.167)
##
## monthmay                             -0.333***
##                                        (0.095)
##
## monthnov                             -0.389***
##                                        (0.141)
```

```
##
## monthoct                                0.183
##                                        (0.181)
##
## monthsep                               0.553***
##                                        (0.211)
##
## day_of_weekmon                         -0.233***
##                                        (0.078)
##
## day_of_weekthu                          0.112
##                                        (0.075)
##
## day_of_weektue                          0.121
##                                        (0.077)
##
## day_of_weekwed                          0.184**
##                                        (0.077)
##
## campaign                               -0.039***
##                                        (0.013)
##
## pdays                                  -0.001***
##                                        (0.0003)
##
## previous                                -0.114
##                                        (0.073)
##
## poutcomenonexistent                    0.469***
##                                        (0.113)
##
## poutcomesuccess                        0.746***
##                                        (0.264)
##
## emp.var.rate                           -1.674***
##                                        (0.159)
##
## cons.price.idx                         2.343***
##                                        (0.288)
##
## cons.conf.idx                           0.022**
##                                        (0.009)
##
## euribor3m                               0.209
##                                        (0.157)
##
## nr.employed                             0.009**
##                                        (0.004)
##
## Constant                               -265.082***
##                                        (44.076)
##
## -------------------------------------------------------
## Observations                           21,342
```

```
## Log Likelihood                        -6,214.144
## Akaike Inf. Crit.                     12,522.290
## ========================================================
## Note:                          *p<0.1; **p<0.05; ***p<0.01
```

The results show that job type and education level are not significant, as well as holding a personal loan or housing loan with the bank. On the other hand, contacting by telephone, month of call, and number of calls during the campaign were significant.

Logit Model (Restricted Model)

```
RESTRICTED_MODEL <- glm(SUBSCRIBED~contact+month+campaign+pdays+poutcome
+emp.var.rate+cons.price.idx+cons.conf.idx,family=binomial(), data = TRAINING_SET)

stargazer(RESTRICTED_MODEL, type = "text")
```

```
##
## ================================================
##                         Dependent variable:
##                     ---------------------------
##                              SUBSCRIBED
## ------------------------------------------------
## contacttelephone             -0.567***
##                               (0.078)
##
## monthaug                      0.307**
##                               (0.119)
##
## monthdec                      0.368*
##                               (0.219)
##
## monthjul                     0.282***
##                               (0.107)
##
## monthjun                     -0.254**
##                               (0.105)
##
## monthmar                     1.387***
##                               (0.132)
##
## monthmay                     -0.499***
##                               (0.085)
##
## monthnov                     -0.272**
##                               (0.108)
##
## monthoct                       0.127
##                               (0.140)
##
## monthsep                       0.179
##                               (0.152)
##
## campaign                     -0.044***
##                               (0.013)
```

```
##
## pdays                          -0.001***
##                                (0.0003)
##
## poutcomenonexistent            0.621***
##                                 (0.074)
##
## poutcomesuccess                0.845***
##                                 (0.251)
##
## emp.var.rate                   -0.816***
##                                 (0.026)
##
## cons.price.idx                 1.184***
##                                 (0.065)
##
## cons.conf.idx                  0.021***
##                                 (0.006)
##
## Constant                      -111.471***
##                                 (6.057)
##
## ------------------------------------------------
## Observations                     21,342
## Log Likelihood                -6,263.849
## Akaike Inf. Crit.             12,563.700
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

Chi-Squared Test of Model Selection

To determine which model will be used on the test set, we will perfrom a chi-squared test of model selection. We will assume a 5% significance level for the statistical test.

```
anova(FULL_MODEL, RESTRICTED_MODEL, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: SUBSCRIBED ~ (age + job + marital + education + default + housing +
##     loan + contact + month + day_of_week + campaign + pdays +
##     previous + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx +
##     euribor3m + nr.employed + true_class) - true_class
## Model 2: SUBSCRIBED ~ contact + month + campaign + pdays + poutcome +
##     emp.var.rate + cons.price.idx + cons.conf.idx
##   Resid. Df Resid. Dev  Df Deviance  Pr(>Chi)
## 1     21295      12428
## 2     21324      12528 -29  -99.409 1.216e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Sine the p-value of 1.216*10^-9 < 0.05, we reject the null hypothesis that both models are the same and select the full model to be used on the test set.

Apply Full Model to Test Set

```
PROBS_TEST_SET<- predict(FULL_MODEL, TEST_SET, type = "response")
# We will use a cutoff probability of 0.5
PREDICTED_CLASS<- as.factor(ifelse(PROBS_TEST_SET>0.50, "accept", "decline"))
TRUE_CLASS<- as.factor(TEST_SET$true_class)
# Generate confusion matrix using caret
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.3.3
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
confusionMatrix(PREDICTED_CLASS, TRUE_CLASS)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction accept decline
##    accept     282     160
##    decline    895    7809
##
##                Accuracy : 0.8846
##                  95% CI : (0.8779, 0.8911)
##     No Information Rate : 0.8713
##     P-Value [Acc > NIR] : 5.944e-05
##
##                   Kappa : 0.2991
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.23959
##             Specificity : 0.97992
##          Pos Pred Value : 0.63801
##          Neg Pred Value : 0.89717
##              Prevalence : 0.12869
##          Detection Rate : 0.03083
##    Detection Prevalence : 0.04833
##       Balanced Accuracy : 0.60976
##
##        'Positive' Class : accept
##
```

We observe that the performance of our full logit model on predicting "accept" is low, as the sensitivity of the model is 23.96%. Overall accuracy is high because the model is good at correctly classifying an individual who will not accept a term deposit subscription. Later on we will compare these results to our rare events techniques.

Generate ROC and AUC

```
library(pROC)
```

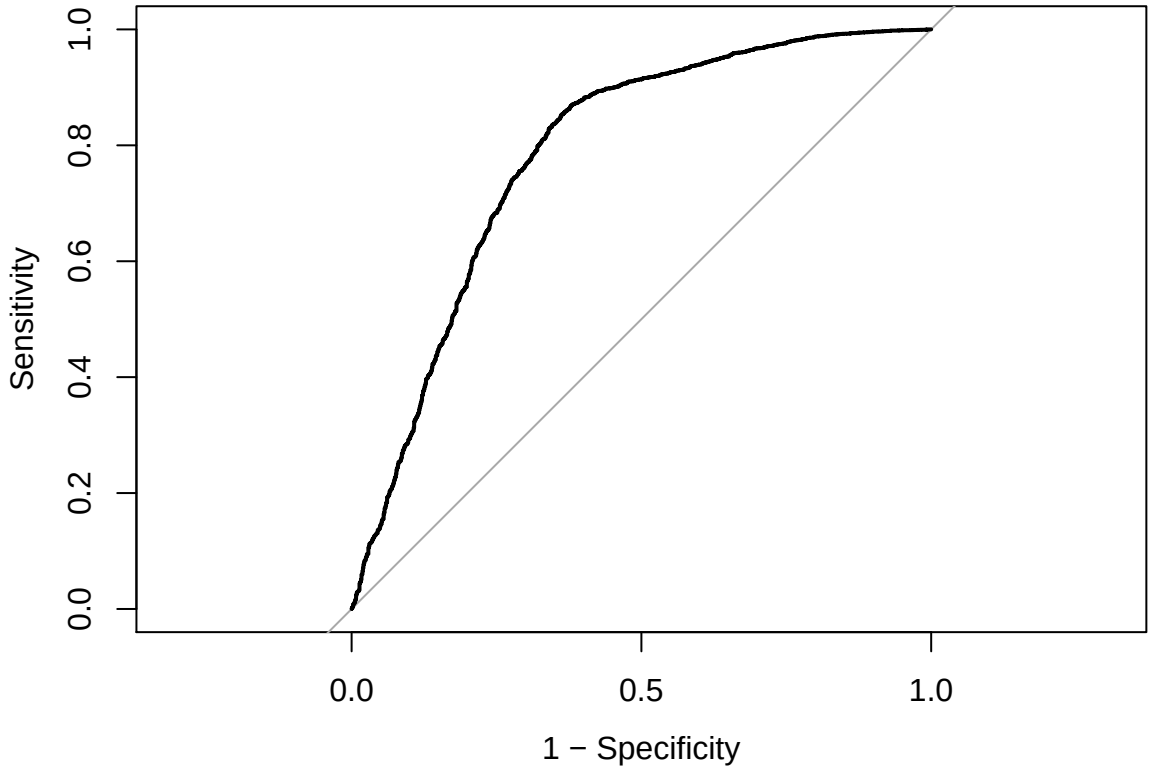
```
## Warning: package 'pROC' was built under R version 3.3.3
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
ROC_CURVE<- roc(TRUE_CLASS~PROBS_TEST_SET)
plot(ROC_CURVE, legacy.axes = TRUE)
```




```
AUC<- auc(ROC_CURVE)
AUC
```

```
## Area under the curve: 0.7852
```

An AUC of 0.7852 means that the full logit model's performance on the test set is "moderate". Next we will implement the Naive Bayes classifier to determine if it performs better at classifying individuals who will accept a term deposit with the bank.

Naive Bayes

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.3.3
```

```
attach(TRAINING_SET)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     SUBSCRIBED
```

```
## The following objects are masked from TRAINING_SET (pos = 9):
##
##     age, campaign, cons.conf.idx, cons.price.idx, contact,
##     day_of_week, default, education, emp.var.rate, euribor3m,
##     housing, job, loan, marital, month, nr.employed, pdays,
##     poutcome, previous, SUBSCRIBED, true_class
```

```
## The following objects are masked from BANK (pos = 11):
##
##     age, campaign, cons.conf.idx, cons.price.idx, contact,
##     day_of_week, default, education, emp.var.rate, euribor3m,
##     housing, job, loan, marital, month, nr.employed, pdays,
##     poutcome, previous, SUBSCRIBED, true_class
```

```
## The following objects are masked from BANK (pos = 12):
##
##     age, campaign, cons.conf.idx, cons.price.idx, contact,
##     day_of_week, default, education, emp.var.rate, euribor3m,
##     housing, job, loan, marital, month, nr.employed, pdays,
##     poutcome, previous
```

```
NAIVE_BAYES<- naiveBayes(true_class~. -SUBSCRIBED, data = TRAINING_SET)
# Apply Naive Bayes model to the test set
PREDICTED_CLASS<- predict(NAIVE_BAYES, TEST_SET)
TRUE_CLASS<- TEST_SET$true_class
# Generate Confusion Matrix
confusionMatrix(PREDICTED_CLASS, TRUE_CLASS)
```

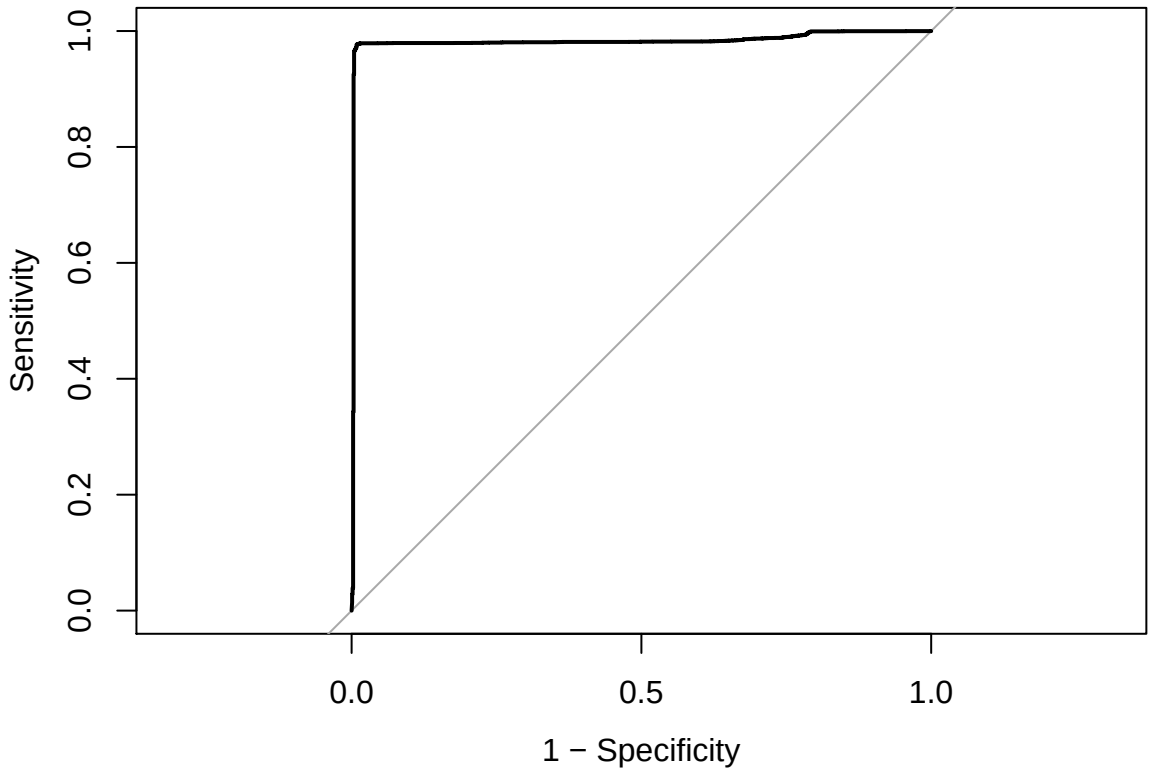
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction accept decline
##    accept    1161     173
##    decline     16    7796
##
##                Accuracy : 0.9793
##                  95% CI : (0.9762, 0.9822)
##     No Information Rate : 0.8713
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9128
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9864
##             Specificity : 0.9783
```

```
##          Pos Pred Value : 0.8703
##          Neg Pred Value : 0.9980
##              Prevalence : 0.1287
##          Detection Rate : 0.1269
##    Detection Prevalence : 0.1459
##       Balanced Accuracy : 0.9823
##
##        'Positive' Class : accept
##
```

We observe that the performance of the Naive Bayes classifier is quite good on the test set. This time we have an accuracy score of 97.93%, with a sensitivity of 98.64% and specificity of 97.83%. This model does well at correctly classifying individuals who will accept a term deposit subscription.

Generate ROC and AUC

```
# Exctract probability values
PROBS<-predict(NAIVE_BAYES, TEST_SET, "raw")
PROBABILITIES<- PROBS[,2]
ROC<-roc(TRUE_CLASS, PROBABILITIES, positive="accept")
plot(ROC, legacy.axes = TRUE)
```




```
AUC<- auc(ROC)
AUC
```

```
## Area under the curve: 0.9826
```

An AUC of 0.9826 means that the Naive Bayes classifier's performance on the test set is very good.

Linear Discriminant Analysis

```
# LDA using mass package
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'TRAINING_SET':
##
##     housing
##
## The following object is masked from 'TRAINING_SET':
##
##     housing
```

```
## The following object is masked from 'BANK':
##
##     housing
##
## The following object is masked from 'BANK':
##
##     housing
```

```
LDA_MODEL<-lda(true_class~. -SUBSCRIBED, data = TRAINING_SET)
```

Apply LDA Model to Test Set

```
LDA_PREDICTIONS<- predict(LDA_MODEL, TEST_SET)
LDA_PREDICTED_CLASS<- LDA_PREDICTIONS$class
POSTERIOR_PROBS<- LDA_PREDICTIONS$posterior[,1]
TRUE_CLASS<- TEST_SET$true_class
# Generate Confusion Matrix
confusionMatrix(LDA_PREDICTED_CLASS, TRUE_CLASS)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction accept decline
##    accept     406     396
##    decline    771    7573
##
##                Accuracy : 0.8724
##                  95% CI : (0.8654, 0.8792)
##     No Information Rate : 0.8713
##     P-Value [Acc > NIR] : 0.3847
##
##                   Kappa : 0.3416
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.34494
##             Specificity : 0.95031
##          Pos Pred Value : 0.50623
##          Neg Pred Value : 0.90760
##              Prevalence : 0.12869
##          Detection Rate : 0.04439
```

```
##    Detection Prevalence : 0.08769
##       Balanced Accuracy : 0.64763
##
##        'Positive' Class : accept
##
```

We observe that the LDA model has an overall accuracy of 87.24%, a sensitivity of 34.5%, and a specificity of 95.03%. This model performed slightly better in terms of sensitivity than the full logit model. Overall the Naive Bayes classifier has performed the best on the test set.

Generate ROC and AUC

```
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 3.3.3
```

```
## Loading required package: gplots
```

```
## Warning: package 'gplots' was built under R version 3.3.3
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(AUC)
```

```
## AUC 0.3.0
```

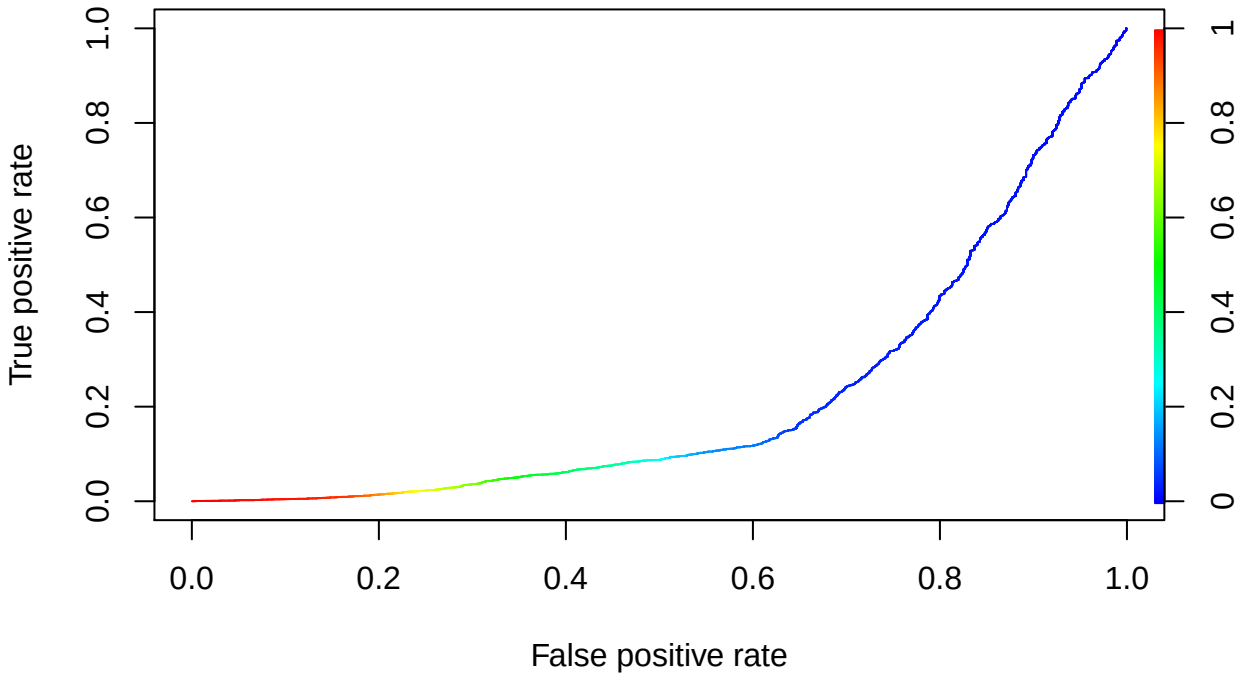
```
## Type AUCNews() to see the change log and ?AUC to get an overview.
```

```
##
## Attaching package: 'AUC'
```

```
## The following objects are masked from 'package:pROC':
##
##     auc, roc
```

```
## The following objects are masked from 'package:caret':
##
##     sensitivity, specificity
```

```
ROC<- prediction(POSTERIOR_PROBS, TRUE_CLASS)
ROC_CURVE<- performance(ROC, "tpr", "fpr")
plot(ROC_CURVE, colorize=TRUE)
```




```
AUC<- auc(roc(POSTERIOR_PROBS, TRUE_CLASS), min=0, max=1)
AUC
```

```
## [1] 0.2186897
```

An AUC of 0.218 means that the performance of the LDA model is very poor. This could be due to the fact that LDA performs much better on smaller datasets.

Modeling Rare Events

To determine the set-up of our training and test set, we must first understand the ratio of declined subscriptions to accepted subscriptions. That ratio is: [26629/3859]= 6.9/1. This means that for every person who accepts a term deposit subscription, 6.9 people will decline a term deposit subscription. If we utilize a 70% training and 30% test split, then the training set must include 2701 accept and 2701 decline observations. The test set must include 1158 accept and 7990 (1158*6.9) decline observations to preserve the balance in the original dataset.

```
# Create an accept dataframe
ACCEPT<- RANDOMIZED_DATA[RANDOMIZED_DATA$true_class == "accept", ]
# Create a decline dataframe
DECLINE<- RANDOMIZED_DATA[RANDOMIZED_DATA$true_class == "decline", ]
```

Now we want to partition these datasets to build our training and test set

```
TRAINING_ACCEPT<- ACCEPT[1:2701, ]
TRAINING_DECLINE<- DECLINE[1:2701, ]
# Put them together
TRAINING_SET<- rbind(TRAINING_ACCEPT, TRAINING_DECLINE)
```

```
# Re-shuffle the training set
TRAINING_SET<- TRAINING_SET[order(runif(5402)), ]
```

Next we do the same for the test set

```
TEST_ACCEPT<- ACCEPT[2702:3859, ]
TEST_DECLINE<- DECLINE[2702:10691, ]
# Put them together
TEST_SET<- rbind(TEST_ACCEPT, TEST_DECLINE)
# Re-shuffle test set
TEST_SET<- TEST_SET[order(runif(9148)), ]
```

Now we can re-perform our statistical analysis on the balanced training set

Logistic Regression on Balanced Training Set

```
attach(TRAINING_SET)
```

```
## The following object is masked _by_ .GlobalEnv:
##
##     SUBSCRIBED
```

```
## The following object is masked from package:MASS:
##
##     housing
```

```
## The following objects are masked from TRAINING_SET (pos = 7):
##
##     age, campaign, cons.conf.idx, cons.price.idx, contact,
##     day_of_week, default, education, emp.var.rate, euribor3m,
##     housing, job, loan, marital, month, nr.employed, pdays,
##     poutcome, previous, SUBSCRIBED, true_class
```

```
## The following objects are masked from TRAINING_SET (pos = 14):
##
##     age, campaign, cons.conf.idx, cons.price.idx, contact,
##     day_of_week, default, education, emp.var.rate, euribor3m,
##     housing, job, loan, marital, month, nr.employed, pdays,
##     poutcome, previous, SUBSCRIBED, true_class
```

```
## The following objects are masked from BANK (pos = 16):
##
##     age, campaign, cons.conf.idx, cons.price.idx, contact,
##     day_of_week, default, education, emp.var.rate, euribor3m,
##     housing, job, loan, marital, month, nr.employed, pdays,
##     poutcome, previous, SUBSCRIBED, true_class
```

```
## The following objects are masked from BANK (pos = 17):
##
##     age, campaign, cons.conf.idx, cons.price.idx, contact,
##     day_of_week, default, education, emp.var.rate, euribor3m,
##     housing, job, loan, marital, month, nr.employed, pdays,
##     poutcome, previous
```

```
# Illusrate the class balance
table(true_class)
```

```
## true_class
```

```
##  accept decline
##    2701    2701
```

```
# Create full model
FULL_MODEL <- glm(SUBSCRIBED~. -true_class, family=binomial(), data = TRAINING_SET)

# Examine results
# stargazer(FULL_MODEL, type = "text")
# I have left out the results to save page space
# The estimates are very similar to the unbalanced training set
```

Apply Full Logit Model to Test Set

```
PROBS_TEST_SET<- predict(FULL_MODEL, TEST_SET, type = "response")
# We will again utilize a cutoff probability of 0.5
PREDICTED_CLASS<- as.factor(ifelse(PROBS_TEST_SET>0.50, "accept", "decline"))
TRUE_CLASS<- as.factor(TEST_SET$true_class)
# Generate confusion matrix
confusionMatrix(PREDICTED_CLASS, TRUE_CLASS)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction accept decline
##    accept     753    1326
##    decline    405    6664
##
##                Accuracy : 0.8108
##                  95% CI : (0.8026, 0.8188)
##     No Information Rate : 0.8734
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3614
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.65026
##             Specificity : 0.83404
##          Pos Pred Value : 0.36219
##          Neg Pred Value : 0.94271
##              Prevalence : 0.12659
##          Detection Rate : 0.08231
##    Detection Prevalence : 0.22726
##       Balanced Accuracy : 0.74215
##
##        'Positive' Class : accept
##
```

Compared to the full logit model on the unbalanced training set, we notice that we get considerable lift in sensitivity. This model is better at classifying true positives than the full logit model that was trained on the unbalanced training set.

Generate ROC and AUC

```
#ROC_CURVE<- roc(TRUE_CLASS~PROBS_TEST_SET)
#plot(ROC_CURVE, legacy.axes = TRUE)
```

```
#AUC<- auc(ROC_CURVE)
#AUC
```

I am getting the error: Error in levels(labels) : argument "labels" is missing, with no default. I will skip generating an ROC and AUC for now.

Naive Bayes on Balanced Training Set

```
NAIVE_BAYES<- naiveBayes(true_class~. -SUBSCRIBED, data = TRAINING_SET)
# Apply Naive Bayes model to the test set
PREDICTED_CLASS<- predict(NAIVE_BAYES, TEST_SET)
TRUE_CLASS<- TEST_SET$true_class
# Generate Confusion Matrix
confusionMatrix(PREDICTED_CLASS, TRUE_CLASS)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction accept decline
##    accept    1147     188
##    decline     11    7802
##
##                Accuracy : 0.9782
##                  95% CI : (0.975, 0.9811)
##     No Information Rate : 0.8734
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9077
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9905
##             Specificity : 0.9765
##          Pos Pred Value : 0.8592
##          Neg Pred Value : 0.9986
##              Prevalence : 0.1266
##          Detection Rate : 0.1254
##    Detection Prevalence : 0.1459
##       Balanced Accuracy : 0.9835
##
##        'Positive' Class : accept
##
```

This model performs relatively similar to the Naive Bayes model that was trained on an unbalanced training set. We see some lift in sensitivity, but at the cost of a lower specificity.

LDA on Balanced Training Set

```
LDA_MODEL<-lda(true_class~. -SUBSCRIBED, data = TRAINING_SET)
# Apply LDA model to test set
LDA_PREDICTIONS<- predict(LDA_MODEL, TEST_SET)
LDA_PREDICTED_CLASS<- LDA_PREDICTIONS$class
POSTERIOR_PROBS<- LDA_PREDICTIONS$posterior[,1]
TRUE_CLASS<- TEST_SET$true_class
# Generate Confusion Matrix
```

```
confusionMatrix(LDA_PREDICTED_CLASS, TRUE_CLASS)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction accept decline
##    accept     751    1338
##    decline    407    6652
##
##                Accuracy : 0.8092
##                  95% CI : (0.801, 0.8173)
##     No Information Rate : 0.8734
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.358
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.64853
##             Specificity : 0.83254
##          Pos Pred Value : 0.35950
##          Neg Pred Value : 0.94234
##              Prevalence : 0.12659
##          Detection Rate : 0.08209
##    Detection Prevalence : 0.22836
##       Balanced Accuracy : 0.74054
##
##        'Positive' Class : accept
##
```

Results of Analysis

Model	Accuracy	Sensitivity	Specificity	Kappa
Full Logit Model	88.46%	23.96%	97.99%	0.2991
Naive Bayes	97.93%	98.64%	97.83%	0.9128
LDA Model	87.24%	34.49%	95.03%	0.3416

Results of Analysis using Modeling Rare Events Techniques

Model	Accuracy	Sensitivity	Specificity	Kappa
Full Logit Model	81.08%	65.02%	83.40%	0.3614
Naive Bayes	97.82%	99.05%	97.65%	0.9077
LDA Model	80.92%	64.85%	83.25%	0.3580

The above tables show the performance of our classification models on the test set. In general, the Naive Bayes classifier performs better than both the full logit model and LDA model. When we applied modeling rare events techniques, we observed some differences in the models' ability to correctly classify if an individual will accept a term deposit subscription. There was a good amount of lift in the true positive rates when trained on the balanced training set. This is what we would expect given that the ratio of accept and decline in the balanced training set is 50-50. A tradeoff of this approach is that the model will be somewhat weaker at

classifying if an individual will decline a term deposit subscription. Since we are most interested in predicting if a customer will accept a term deposit subscription, the sensitivity measure has more weight in determining an optimal classifier. One thing to note is that the sensitivity and specificity of the Naive Bayes model are very close to 1, which raises our suspicions of over-fitting. Overall it seems that the Naive Bayes model would be a good candidate to be applied in a decision support system. In regards to the company deciding to use rare events techniques, one must decide if 12.66% of a sample is considered a rare event. Other literature suggests that an event that occurs less than 5% of the time is truly a rare event. In the next section I will discuss the conclusions of the data mining project.

Conclusion

The goal of the data mining project was to determine if a machine learning model could accurately predict if an individual would accept or deny a term deposit subscription with the bank. If the model(s) performed well enough, then they could be potentially used in a decision support system to help telemarketers of the bank better target customers who have a higher probability of accepting a term deposit subscription. Given the results of the statistical analysis, we can reasonably conclude that the Naive Bayes classifier is a great option for the bank to deploy. The Naive Bayes model has all the characteristics of a well performing classifier: high accuracy, high true positive rate, high true negative rate, and high AUC. In our analysis we considered applying modeling rare events techniques given that the prior probability of accepting a term deposit subscription was 12.65%. Whether or not this percentage represents a "rare event" or not is up for debate. Applying such techniques led to considerable lift in the sensitivity measure for the full logit model and LDA model. Overall the model that we would suggest be deployed or tested on new data would be the Naive Bayes model. The next steps of the data mining project would be to deploy the model on new data to see if it performs well at correctly classifying new customers.