Bank Marketing Case Study

Alex Martinez, Josh Gardner, Cameron Playle, and Guillermo Gallardo

2024-09-22

Executive Summary:

Our best model was the logistic regression model, which identified **age, poutcome, campaign, previous, cons.conf.idx, contact, cons.price.idx, and emp.var.rate** as the most significant predictors for our response variable.

The two most significant variables were **emp.var.rate** and **cons.price.idx**. The model achieved an overall **accuracy of 73%**, with a **sensitivity of 77%** and a **specificity of 70%.**

Macroeconomic factors such as the employment variation rate and the 3-month Euribor rate influence subscription decisions. A positive employment outlook tends to increase the likelihood of subscriptions.

Our Problem:

The primary problem this case study addresses is identifying the variables that significantly influence whether customers will subscribe ('yes') or not ('no') to a bank term deposit product. From a theoretical perspective, the study aims to determine which predictor variables significantly impact the likelihood of a customer subscribing to a bank term deposit, using statistical models such as logistic and linear regression to analyze the data

Literature Review:

Existing Works in Theoretical and Application Realm Research in customer retention and subscription prediction in financial services has been extensive. Theoretical studies such as Predictive Modeling for Marketing Campaigns by Chapman et al. (2015) highlight the utility of machine learning in optimizing campaign performance. Studies like Moro et al. (2014) applied data mining techniques on a similar Portuguese banking dataset, showing that models like logistic regression and decision trees provide high accuracy in predicting term deposit subscriptions.

Further, works such as Customer Analytics in Financial Services by Homburg et al. (2016) validate the impact of socioeconomic variables like employment rates and consumer confidence on consumer financial decisions. These existing works provide a solid foundation for applying predictive models in this case study.

By leveraging these insights, this project aims to contribute to the growing body of research that helps financial institutions fine-tune their marketing strategies through data-driven approaches.

Methods:

Type of Variables: When loading the data, the majority of the variables were imported as character variables. We had to adjust them to match the information given to us. After making the necessary changes, we ended up with factors, numeric, and integer variables. We also created dummy variables, which we will discuss later.

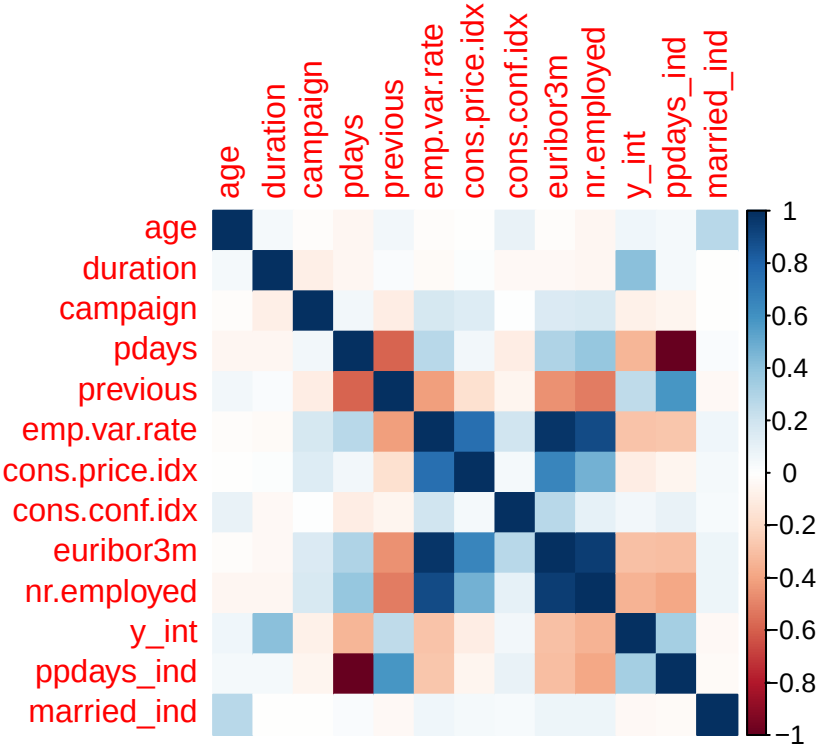
Sample Size and Techniques: The sample size is 4,119 with 21 variables. Due to the sample being unbalanced towards a "no" answer on the y variable, we resampled the data. This process reduced the 4,119 samples to slightly over 900, which we used to train our model.

Models, Assumptions, and Limitations: During this case study, we used two models: a linear model and a logistic model. The assumptions for the linear model include linearity, independence, homoscedasticity, and no multicollinearity. For the logistic model, the requirements are a binary dependent variable, independence of observations, linearity of the logit, no multicollinearity, and no perfect separation.

Data:

The dataset provided is well-structured, containing 21 variables related to client demographics and financial information, with a total of 4,119 observations. Although there are no *NA* values, we did notice a few columns containing 'Unknown' entries. In terms of variable types, we treated some as factors and others as numeric, as many of the original variables were labeled as characters. For instance, we converted *education* from a character to a factor and *emp.var.rate* from a character to numeric.

Upon checking for multicollinearity, we identified that only one pair of variables—pdays and duration—exceeded our predefined collinearity threshold. Given this multicollinearity, along with other considerations regarding the duration variable, we made the decision to exclude duration from the final model.



Our dataset was unbalanced, with 3,668 records labeled as *No* and only 451 labeled as *Yes* in the y variable. To address this imbalance, we created a balanced dataset by splitting the data between the *Yes* and *No* labels and then sampling from each group. This resulted in a new dataset called *df_bal*, which contains an equal number of records. Balancing the data will help us train our model more effectively by ensuring that it doesn't become biased towards the majority class. This should improve the model's ability to accurately predict both outcomes and perform well across various metrics.

Unbalanced Dataset

```
##   no  yes
## 3668  451
```

Balanced Dataset

```
##  no yes
## 451 451
```

For our final dataset, we selected 19 variables initially. After applying backward elimination, we narrowed it down to 7 key variables for our final model. The variables we retained are: age, poutcome, campaign, cons.conf.idx, contact, cons.price.idx, and emp.var.rate.

Results:

Here we will be discussing the outputs from our two models we applied to the data set to best predict if a client will subscribe (yes/no) to a term deposit (y). We first begin with a simple linear model and use the backwards selection method to keep only the significant variables. Here is the output:

```
summary(lmf)
```

```
##
## Call:
## lm(formula = y_int ~ age + contact + campaign + poutcome + emp.var.rate +
##     cons.price.idx + cons.conf.idx, data = df_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.03746 -0.32677 -0.06865  0.34689  0.90629
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -27.807890   3.836511  -7.248 1.10e-12 ***
## age                   0.002718   0.001380   1.969 0.049333 *
## contacttelephone     -0.165558   0.043288  -3.825 0.000142 ***
## campaign             -0.017232   0.006093  -2.828 0.004813 **
## poutcomenonexistent   0.099466   0.053596   1.856 0.063887 .
## poutcomesuccess       0.168443   0.069832   2.412 0.016111 *
## emp.var.rate         -0.175603   0.015972 -10.994  < 2e-16 ***
## cons.price.idx        0.305554   0.041521   7.359 5.12e-13 ***
## cons.conf.idx         0.011460   0.003388   3.383 0.000757 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.4273 on 712 degrees of freedom
## Multiple R-squared:  0.2787, Adjusted R-squared:  0.2706
## F-statistic:  34.4 on 8 and 712 DF,  p-value: < 2.2e-16
```

We see that the model kept age, contact, campaign, poutcome, emp.var.rate, cons.price.idx, and cons.conf.idx. Right away, we see that the model is significant but the Adjusted R-Square is pretty low at 27%. This is expected since our dependent variable is binary and a Linear regression model is not the best option for binary outcomes.

With this conclusion we move on to a Logistic regression model since this gives us a prediction for our dependent variable. This will help us predict if a client will subscribe to a term deposit (indicated as 1).

The final model that is created using the backwards selection method is the following:

```
summary(glmf)
```

```
##
## Call:
## glm(formula = y_int ~ age + contact + campaign + previous + poutcome +
##     emp.var.rate + cons.price.idx + cons.conf.idx, family = binomial,
##     data = df_train)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.687e+02  2.861e+01  -5.896 3.72e-09 ***
## age                  1.703e-02  8.322e-03   2.047 0.040669 *
## contacttelephone    -9.801e-01  2.562e-01  -3.825 0.000131 ***
## campaign            -1.317e-01  4.858e-02  -2.711 0.006702 **
## previous             6.681e-01  4.938e-01   1.353 0.176123
## poutcomenonexistent  1.307e+00  6.334e-01   2.064 0.039024 *
## poutcomesuccess      1.304e+00  5.235e-01   2.492 0.012719 *
## emp.var.rate        -8.974e-01  1.069e-01  -8.395  < 2e-16 ***
## cons.price.idx       1.812e+00  3.092e-01   5.859 4.65e-09 ***
## cons.conf.idx        5.959e-02  1.981e-02   3.009 0.002623 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 999.48  on 720  degrees of freedom
## Residual deviance: 763.35  on 711  degrees of freedom
## AIC: 783.35
##
## Number of Fisher Scoring iterations: 6
```

Before we get into the confusion matrix. Lets explain the relationship of each independent variable to the dependent variable. We first compute the exponential of our coefficient ratio to get the odds ratio.

```
##     age
## 1.01718
```

```
## contacttelephone
##        0.3752893
```

```
## <NA>
##   NA

## <NA>
##   NA

## <NA>
##   NA

## <NA>
##   NA

## emp.var.rate
##    0.4076233

## cons.price.idx
##        6.12036

## cons.conf.idx
##      1.061404
```

An additional year of age increases the odds of a client subscribing to a term deposit by a factor of 1.02. Key findings from the model include:

- Age: Each additional year increases the odds by 1.02.
- contactelephone: Clients contacted via telephone have reduced odds, decreasing by a factor of 0.38.
- emp.var.rate: A higher rate decreases the odds by 0.41.
- cons.price.idx: An increase significantly boosts the odds by 6.12.
- cons.conf.idx: Each increase raises the odds by 1.06.

Using this model on our test sample, we then evaluated its performance and identified the optimal cutoff to maximize both specificity and sensitivity.

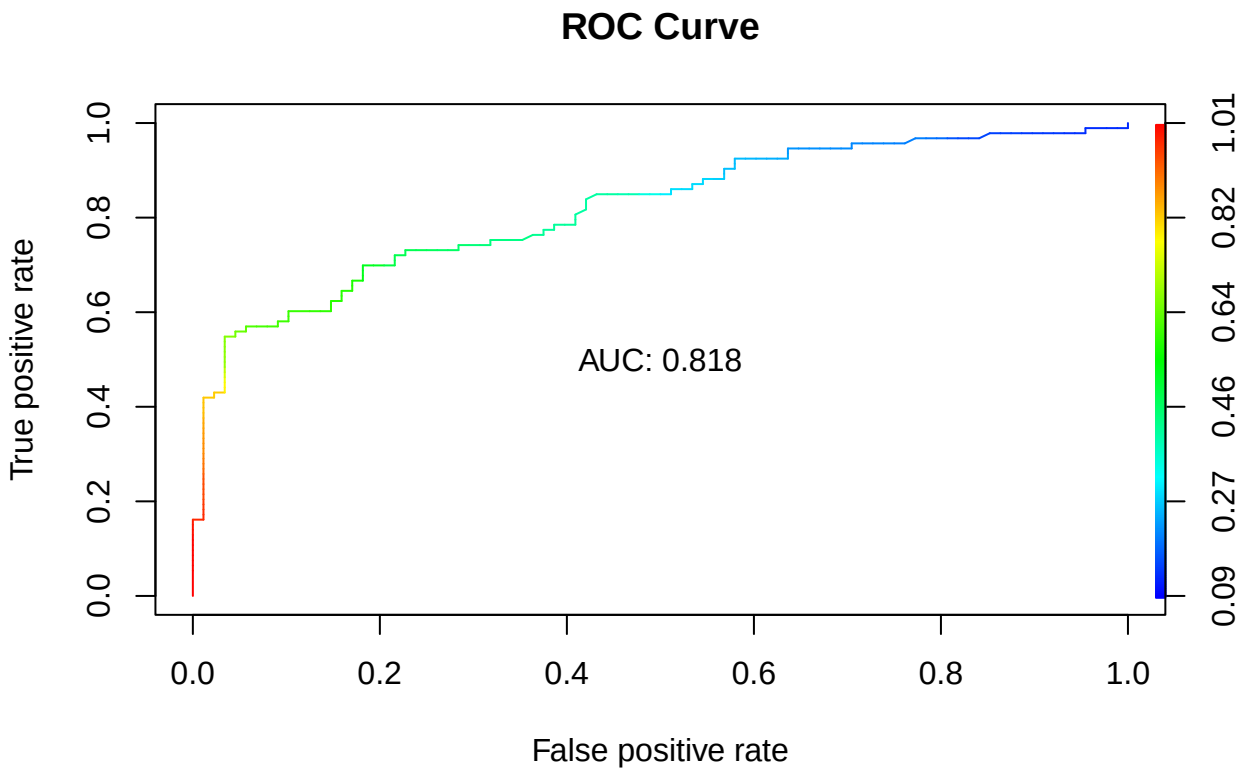
```r
# Predict the probability (p) of
glmftest <- glm(formula = y_int ~ age + contact + campaign + previous + poutcome +
    emp.var.rate + cons.price.idx + cons.conf.idx, family = binomial,
    data = df_test)
probabilities <- predict(glmftest, type = "response")
predicted.classes <- ifelse(probabilities > 0.5, 1, 0)

#### end of default cutoff of 0.5
#### optimal cutoff

#ROC Curve and AUC
pred <- prediction(probabilities,df_test$y_int)
pred
```

```
## A prediction instance
##   with 181 data points
```

```r
#Predicted Probability and True Classification

# area under curve
auc <- round(as.numeric(performance(pred, measure = "auc")@y.values),3)
auc
```

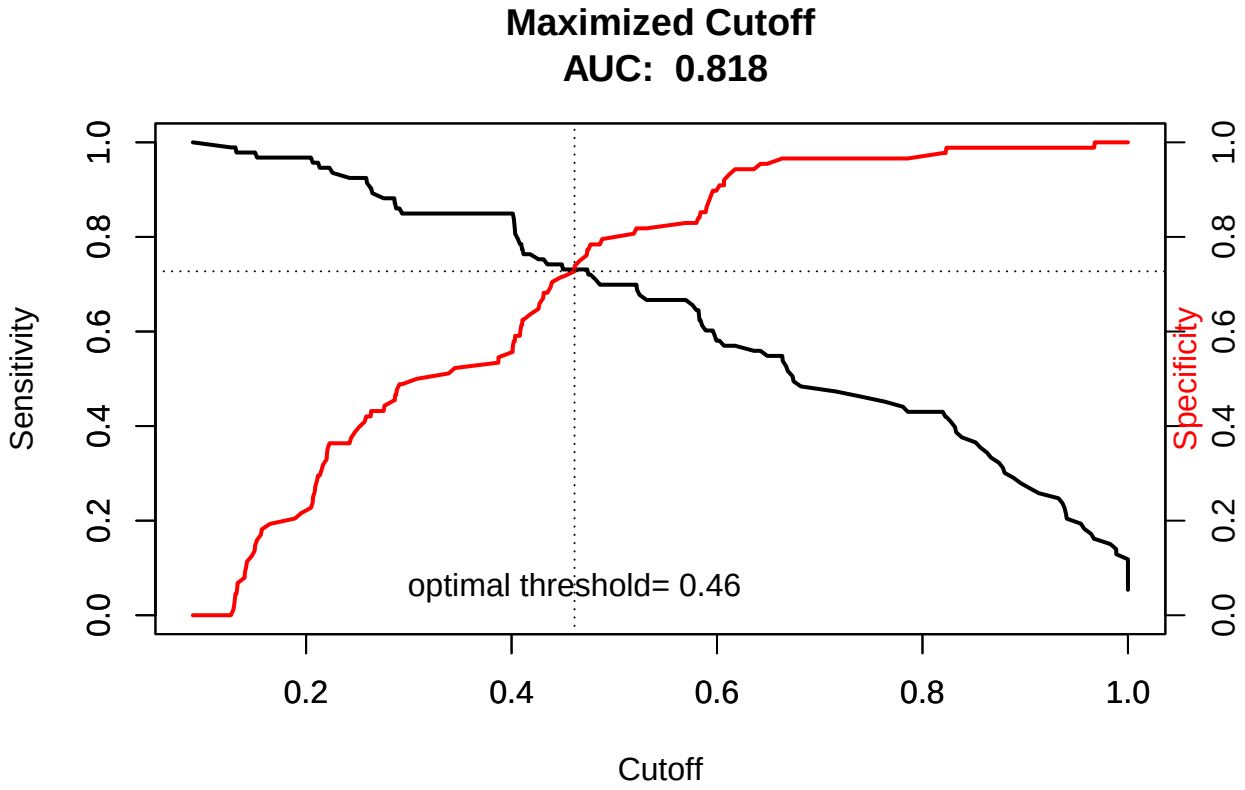
```
## [1] 0.818
```

```r
#plotting the ROC curve and computing AUC
perf <- performance(pred, "tpr","fpr")
plot(perf,colorize = T, main = "ROC Curve")
text(0.5,0.5, paste("AUC:", auc))
```




```r
# computing threshold for cutoff to best trade off sensitivity and specificity
#first sensitivity
plot(unlist(performance(pred, "sens")@x.values), unlist(performance(pred, "sens")@y.values),
     type="l", lwd=2,
     ylab="Sensitivity", xlab="Cutoff", main = paste("Maximized Cutoff\n","AUC: ",auc))

par(new=TRUE) # plot another line in same plot

#second specificity
plot(unlist(performance(pred, "spec")@x.values), unlist(performance(pred, "spec")@y.values),
     type="l", lwd=2, col='red', ylab="", xlab="")
axis(4, at=seq(0,1,0.2)) #specificity axis labels
```

```r
mtext("Specificity",side=4, col='red')

#find where the lines intersect
min.diff <-which.min(abs(unlist(performance(pred, "sens")@y.values) - unlist(performance(pred, "spec")@y
min.x<-unlist(performance(pred, "sens")@x.values)[min.diff]
min.y<-unlist(performance(pred, "spec")@y.values)[min.diff]
optimal <-min.x #this is the optimal points to best trade off sensitivity and specificity

abline(h = min.y, lty = 3)
abline(v = min.x, lty = 3)
text(min.x,0,paste("optimal threshold=",round(optimal,2)), pos = 3)
```




```r
##OPTIMAL CUTOFF FOR BEST SENSITIVITY and specificity
#create Prediction Indicators for y
df_test$Pred_Y_best <- ifelse(df_test$PredProb >= 0.46, 1, 0)
caret::confusionMatrix(as.factor(df_test$y_int),as.factor(df_test$Pred_Y_best), positive = '1') #this f
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 69 19
##          1 30 63
##
##                Accuracy : 0.7293
```

```
##                  95% CI : (0.6584, 0.7925)
##     No Information Rate : 0.547
##     P-Value [Acc > NIR] : 3.447e-07
##
##                   Kappa : 0.46
##
##  Mcnemar's Test P-Value : 0.1531
##
##             Sensitivity : 0.7683
##             Specificity : 0.6970
##          Pos Pred Value : 0.6774
##          Neg Pred Value : 0.7841
##              Prevalence : 0.4530
##          Detection Rate : 0.3481
##    Detection Prevalence : 0.5138
##       Balanced Accuracy : 0.7326
##
##        'Positive' Class : 1
##
```

```
# accuracy: 73%
# sens: 77%
# Spec: 70%
```

We see the best cutoff for the highest sensitivity and specificity is at 0.46. With this, the performance of our model on the test data set is 73% accurate overall, and our sensitivity is 77% with a specificity of 70%

Conclusion

Based on our analysis, the logistic regression model (GLM) outperformed the linear regression model (LM) in predicting whether a client will subscribe to a term deposit. The GLM achieved higher accuracy with strong sensitivity, specificity, and an AUC of 0.818. It effectively handled the binary nature of the response variable and identified significant predictors such as age, contact type, campaign history, previous outcome, employment variation rate, and consumer price and confidence indexes. In comparison, the LM struggled with the binary outcome, showing poor fit since linear regression is designed for continuous variables. Therefore, the GLM is the best option between the two.

For future improvements, exploring methods like Random Forests could enhance model performance. Random Forests handle the binary nature of the data well, capturing non-linear relationships and interactions. By averaging multiple decision trees, they reduce overfitting and improve prediction accuracy, while also identifying the most important predictors for client subscriptions.

The key insight from the analysis is that for every increase in the consumer price index (CPI), customers are 6.12 times more likely to subscribe to term deposits. Based on this finding, the bank should enhance its outreach efforts during periods of rising CPI to maximize subscriptions.