BookBinders Book Club

Visha Arumugam(vcu526), Michael Grogan(ldl776),Sanyogita Apte(jlh562)

September 28, 2021

I - Executive Summary

We recommend that the Bookbinders Book Club begin to applying logistic regression to their customer sales data in order to maximize profit when mailing brochures to larger markets. We found that by training models on synthetically balanced purchase data, we can improve profitability by over 86% vs the scenario where the company does not target their brochures.

II - The Problem

The Bookbinders Book Club is a specialty book distributor that is seeking to survive in an business environment increasingly dominated by superstores like Amazon that are able to leverage economies of scale to out-compete book clubs and smaller retail stores. In order to be more competitive, BBBC has collected data on its customers and plans to use that data to identify the characteristics of the individuals that are most likely to buy a book when mailed a specialty brochure.

The data they have on their customers is primarily numeric data relating to how many books the customer has bought from different categories such as Cooking, Art, etc.

They have specified that they want to find the most useful model out of the following options: logistic regression, linear regression, and support vector machine.

Ultimately, BBBC would like to see the potential profit from mailing brochures to their Midwest client-base of 50,000 using a targeted model compared with sending a brochure to the entire population.

We will show how we evaluated the three models and determined the top performer in terms of potential profit for their Midwest market.

III - Review of Related Literature

There were very few Marketing Analysis happened with this BookBinders Book club data set and based upon the usage of various tools and technologies,various exploration and various prediction techniques, different analysis came with different conclusion and Recommendation.

Few of the Analysis examples on this data set uses the choice based analysis , which will evaluate the effectiveness of marketing efforts based on past purchase data at rather low costs. As per the choice based analysis, it is concluded that every person who has purchased a product within the past four months will be considered a good target in order to promote the direct marketing campaign for the book “The Art History of Florence”.

Some analysis has been conducted using RFM analysis (which is a marketing technique used to quantitatively rank and group customers based on the recency, frequency and monetary total of their recent transactions to identify the best customers and perform targeted marketing campaigns.) and binary logistic regression in order to promote the direct marketing campaign for the book “The Art History of Florence”.

IV - Methodology

We are going to use three prediction algorithm techniques to identify the potential buyer of the book through direct mail brochure. The three models are as follows Linear Regression, Logistic Regression and Support Vector Machine.

To simplify and improve our models, we would eliminate predictors from the model that are not significant. However when performing exploratory modeling of the training data, most of the variables were shown to be significant. The exception is that the inclusion of the ‘First_purchase’ variable results in a failure to converge for the logistic model. Upon further examination, there is a statisticall significant difference in means between the buying and nonbuying customers this t test shows:

```
t.test(bbtrain$First_purchase[bbtrain$Choice==0],bbtrain$First_purchase[bbtrain$Choice==1])
```

```
## 
##  Welch Two Sample t-test
## 
## data:  bbtrain$First_purchase[bbtrain$Choice == 0] and bbtrain$First_purchase[bbtrain$Choice
== 1]
## t = -0.11997, df = 630.04, p-value = 0.9045
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.055223  1.818557
## sample estimates:
## mean of x mean of y 
##  22.54667  22.66500
```

However the difference between the means is so small as to be useless in terms of prediction, because the distribution of the variable is much wider.

```
table(bbtrain$First_purchase)
```

```
## 
##   2   4   6   8  10  12  14  16  18  20  22  24  26  28  30  32  34  36  38  40 
##  55  58  66  99 113 126 115 143 100  70  71  54  39  45  57  47  34  33  40  27 
##  42  44  46  48  50  52  54  56  58  60  62  64  66  68  70  72  74  76  78  80 
##  16  25  19  14  20  10  13   7  10  11  11   3  17   8   5   4   2   1   4   1 
##  82  84  86  96 
##   3   2   1   1
```

As a result, this variable is excluded from the models.

The data then needs to be balanced for the target class, because with the unmodified data set a classifier could achieve high accuracies by depending on the population bias in the sample. A balanced training set is created by resampling the “yes” observations to match the quantity of “no” observations. However, after the classifier is trained, it will be tested on the unbalanced test set in order to determine how the classifier would perform under real conditions.

We choose to test all three models on three different sets of training data with different methods of resampling. The first with unbalanced, unaltered data. The second training set resamples the “buy” observations so that they equal the “nonbuy” observations so they end up equal. The third set uses ROSE (Random Oversampling

Examples)

Following is a brief summary of the classifiers we used:

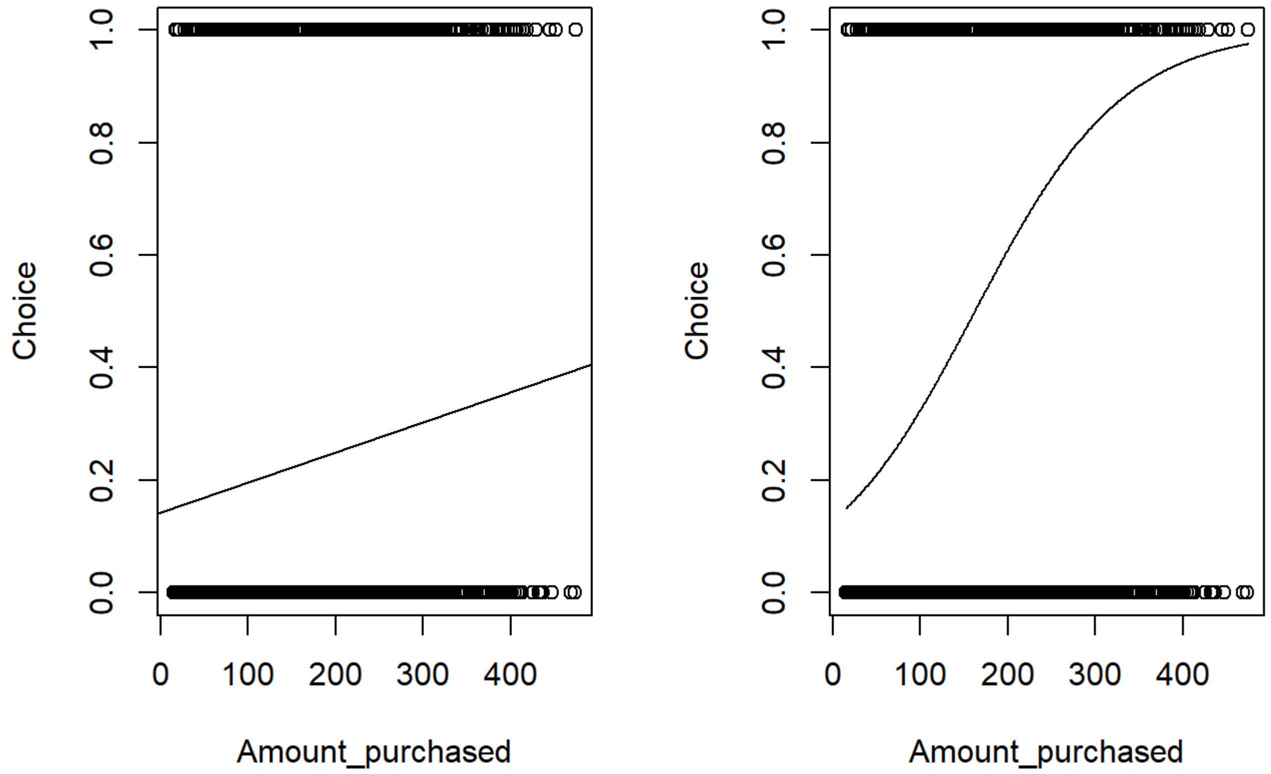
Linear Regression: Linear regression attempts to model the relationship between dependent and independent variables by fitting a linear equation to observed data. The most common method for fitting a regression line is the method of least-squares. This method calculates the best-fitting line for the observed data by minimizing the sum of the squares of the vertical deviations from each data point to the line.

Logistic Regression: Logistic Regression is a parametric classification method in which is used to model the probability of a certain class or event existing based upon the independent variables.In Logistic Regression, we don’t directly fit a straight line to our data like in linear regression. Instead, we fit a S shaped curve, called Sigmoid, to our observations.

Support Vector Machines: SVM is a learning algorithm used in regression tasks. However, SVM is preferable in classification tasks. This algorithm is based on the following idea: if a classifier is effective in separating convergent non-linearly separable data points, then it should perform well on dispersed ones. SVM finds the best separating line that maximizes the distance between the hyperplanes of decision boundaries.

Because the target variable (Choice) is a binary variable, it doesn’t lend itself well to linear regression. And yet, we can still form a version of classification by rounding the predicted value of Choice to the nearest integer. This is not equivalent to the probabilities produced by the logistic regression, but we can use the same technique for assigning the prediction a label of 1 or 0.

Compare below the linear regression of Amount_purchased vs the logistic regression



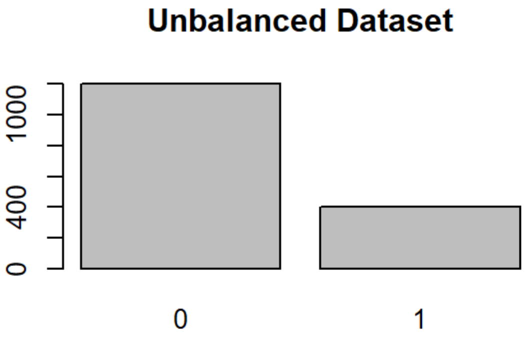
V - Data

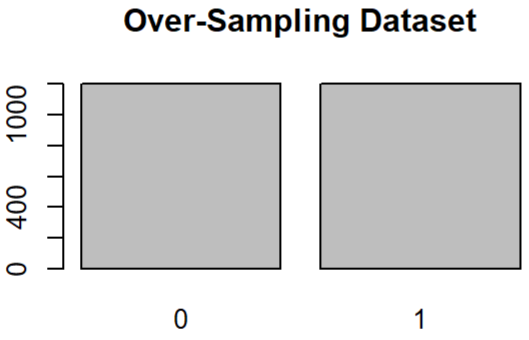
The dataset to be used is the sample of Bookbinders Book Club customers from Pennsylvania, New York, and Ohio Which contains the details of whether the customers are willing to buy the book “The Art of Florescence” or not through direct mailing the brochure.

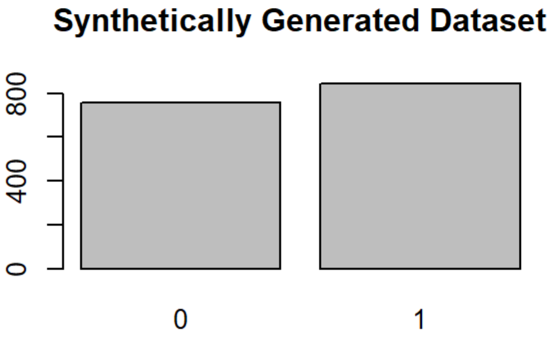
Along the 1600 records in the dataset, 400 members who bought the book and 1200 who didn’t bought the book, which ends up in a imbalanced dataset. As part of this Case study we have tried a Oversampling and Synthetically Data Generation sampling in order to increase the prediction of customers who will buy the book.

Oversampling: It replicates the observations from minority class to balance the data. An advantage of using this method is that it leads to no information loss. The disadvantage of using this method is that, since oversampling simply adds replicated observations in original data set, it ends up adding multiple observations of several types, thus leading to overfitting.

Synthetic Data Generation: Instead of replicating and adding the observations from the minority class, it overcome imbalances by generates artificial data based on feature space (rather than data space) similarities from minority samples. It is also a type of oversampling technique.


Unbalanced Dataset


Over-Sampling Dataset


Synthetically Generated Dataset

VI - Findings

```
## [1] "Logit Unbalanced"
```

```
##           Reference
## Prediction    1    0
##          1   77  123
##          0  127 1973
## [1] "Logit Overbalanced"
```

```
##           Reference
## Prediction    1    0
##          1  142  499
##          0   62 1597
```

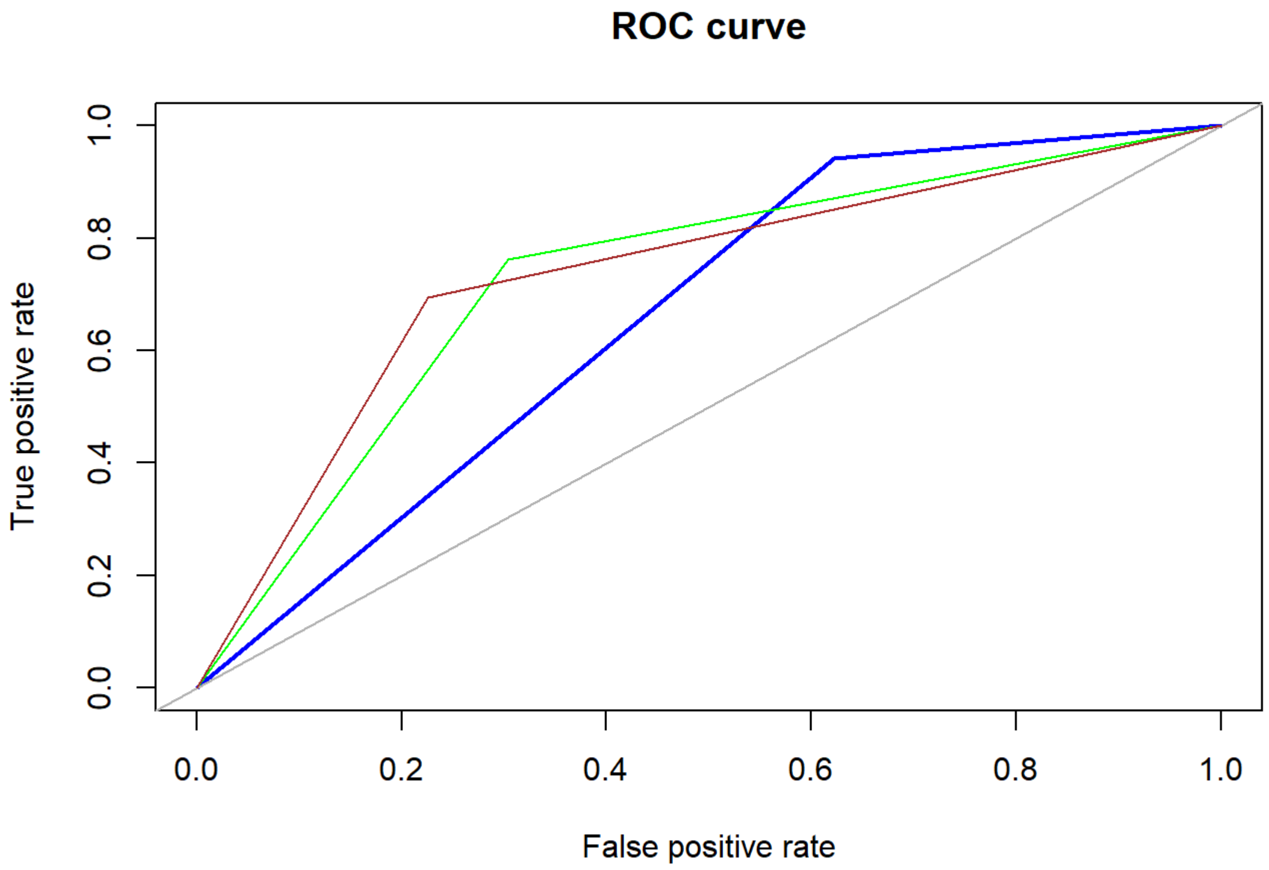
```
## [1] "Logit Synthetic balanced"
```

```
##           Reference
## Prediction    1    0
##          1  158  640
##          0   46 1456
```

```
## [1] "Logit ROC Curves"
```

```
## Area under the curve (AUC): 0.659
```

```
## Area under the curve (AUC): 0.729
```



```
## Area under the curve (AUC): 0.735
## [1] "SVM Confusion Matrices"
```

```
## [1] "SVM Unbalanced"
```

```
##           Reference
## Prediction    1    0
##          1   56   81
##          0  148 2015
```

```
## [1] "SVM Overbalanced"
```

```
##           Reference
## Prediction    1    0
##          1  139  494
##          0   65 1602
```

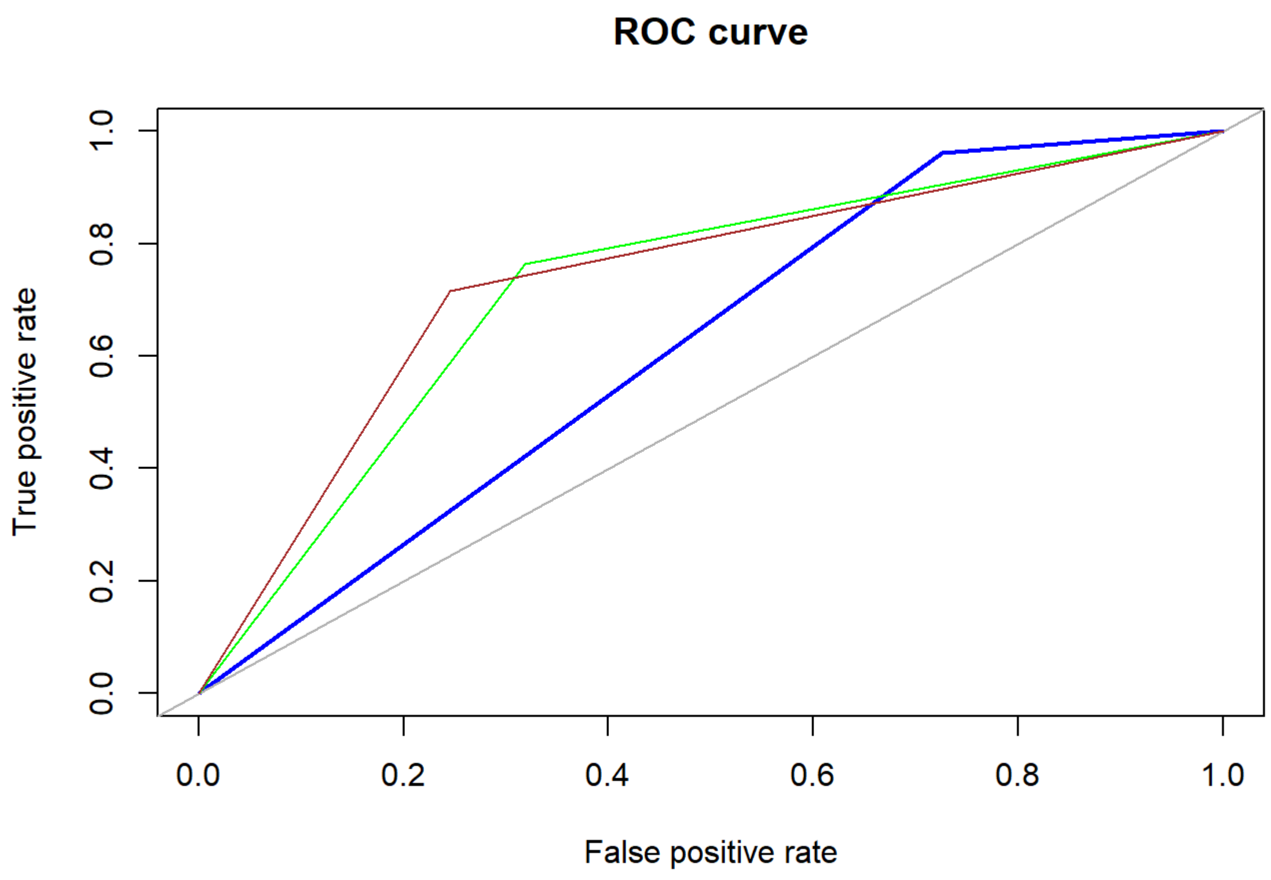
```
## [1] "SVM ROSE balanced"
```

```
##           Reference
## Prediction    1    0
##          1  154  596
##          0   50 1500
```

```
## [1] "SVM ROC Curves"
```

```
## Area under the curve (AUC): 0.618
```

```
## Area under the curve (AUC): 0.723
```



```
## Area under the curve (AUC): 0.735
```

```
## [1] "Linear Model Confusion Matrix"
```

```
## [1] "Linear Unbalanced"
```

```
##              Reference
## Prediction    1    0
##            1   60   89
##            0  144 2007
```

```
## [1] "Linear Overbalanced"
```

```
##              Reference
## Prediction    1    0
##            1  142  487
##            0   62 1609
```

```
## [1] "Linear Synthetic balanced"
##           Reference
## Prediction    1    0
##          1  158  638
##          0   46 1458
```

We calculate the performance of the model in terms of profitability by first determining the profitability of the scenario with no discrimination as to who receives a brochure.

If the population of 50,000 Midwest customers has the same fraction of their population as buyers of BBBC books (8.8%), then the income potential from those buyers if $45,235 if they are all correctly identified. If all 50,000 customers are sent a brochure, the mailing costs will be $32,500 at $0.65 per brochure. This blanket approach would yield an ultimate profit of $12,735

One by one, we will show how this profitability changes based on the model that is used. First we multiply the population by the detection prevalence of the model, which is the fraction of the population that the model predicts will buy a book if mailed a brochure. Next, we multiply this subset of the population by the positive predictive value, which is the fraction of our predicted buyers that actually turn out to be buyers.

So as opposed to calculating profits and costs based on the entire 50,000 Midwesterners, we calculate them based on the fraction of the population that the model chooses as potential buyers

The list of percentages below details the extent to which each model outperforms or underperforms the profitability of the blanket approach.

```
#cost is .65 per mail sent, book cost is 15 with overhead of 45% of cost, and selling price is 3
1.95
#The following assumes that Midwest will have a similar buying population as the test data


predictions<-list(predlm,predlm.over,predlm.rose,predlog,predlog.over,predlog.rose,predsvm,preds
vm.over,predsvm.rose)
predlabel<-c("Raw Linear Regression","Balanced Linear Regression","Synthetic Linear Regression",
"Raw Logit","Balanced Logit","Synthetic Logit","Raw SVM","Balanced SVM","Synthetic SVM")

mailcost<-0.65
profit<-31.95-(15*1.45)
Midwestbase<-50000
buyerfraction<-sum(bbtest$Choice==1)/length(bbtest$Choice)

blanketprofit<-((Midwestbase*buyerfraction)*profit)-(Midwestbase*mailcost)
```

```
## [1] "Profitability using Model to select mailers, vs mailing every Midwest customer"
```

```
## [1] "Profit mailing everyone: $12734.78"
```

```
for(i in 1:length(predictions)){

bestpredictor<-unlist(predictions[i])

#percentage of buy predictions made by model and percentage of buy predictions that are correct
detectionprevalence<-as.numeric(caret::confusionMatrix(bestpredictor,bbtest$Choice)$byClass[10])
pospredvalue<-as.numeric(caret::confusionMatrix(bestpredictor,bbtest$Choice)$byClass[3])

targetedprofit<-((Midwestbase*detectionprevalence*pospredvalue)*profit)-(Midwestbase*detectionpr
evalence*mailcost)

outperformance<-(targetedprofit-blanketprofit)*100/blanketprofit



print(predlabel[i])
print(paste(round(outperformance, 2), "%", sep=""))
}
```

```
## [1] "Raw Linear Regression"
## [1] "-12.06%"
## [1] "Balanced Linear Regression"
## [1] "77.46%"
## [1] "Synthetic Linear Regression"
## [1] "86.79%"
## [1] "Raw Logit"
## [1] "11.88%"
## [1] "Balanced Logit"
## [1] "76.13%"
## [1] "Synthetic Logit"
## [1] "86.57%"
## [1] "Raw SVM"
## [1] "-17.69%"
## [1] "Balanced SVM"
## [1] "71.79%"
## [1] "Synthetic SVM"
## [1] "84.93%"
```

As we can see, the models trained with the unbalanced dataset yielded profitability much lower than the balanced models (even worse profitability than the no-information approach).

The synthetic ROSE method of equalizing the samples actually results in a slight bias towards positive (buy) observations, which results in models that are slightly more likely to assume that the customer is a potential buyer. Because the cost is so low for the brochures, the increased likelihood of predicting buy due to the imbalanced data yields a higher rate of profitability.

As a result, the models yielding the highest profitability were trained on the ROSE dataset, and of those the model with the highest performance is very close between the logistic and linear regression models.

```
summary(log_rose)
```

```
##
## Call:
## glm(formula = Choice ~ . - First_purchase, family = binomial,
##     data = bbtrain.rose)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4923  -0.9878   0.4033   0.9899   2.2519
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.6734583  0.1568345   4.294 1.75e-05 ***
## Gender           -0.6472371  0.1029992  -6.284 3.30e-10 ***
## Amount_purchased  0.0011993  0.0005485   2.186   0.0288 *
## Frequency        -0.0629266  0.0070785  -8.890  < 2e-16 ***
## Last_purchase     0.1178842  0.0232849   5.063 4.13e-07 ***
## P_Child          -0.2170946  0.0549961  -3.947 7.90e-05 ***
## P_Youth           0.0329491  0.0803911   0.410   0.6819
## P_Cook           -0.2306521  0.0556223  -4.147 3.37e-05 ***
## P_DIY            -0.3658590  0.0787928  -4.643 3.43e-06 ***
## P_Art             0.6946676  0.0708016   9.811  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2213.4  on 1599  degrees of freedom
## Residual deviance: 1869.9  on 1590  degrees of freedom
## AIC: 1889.9
##
## Number of Fisher Scoring iterations: 4
```

```
summary(lm_rose)
```

```
##
## Call:
## glm(formula = Choice ~ . - First_purchase, family = gaussian,
##     data = bbtrain.rose)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.09067  -0.40425   0.03619   0.40875   0.99199
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.6404973  0.0312592  20.490  < 2e-16 ***
## Gender           -0.1361894  0.0204109  -6.672 3.46e-11 ***
## Amount_purchased  0.0002387  0.0001097   2.176   0.0297 *
## Frequency        -0.0127433  0.0013379  -9.525  < 2e-16 ***
## Last_purchase     0.0230043  0.0045502   5.056 4.79e-07 ***
## P_Child          -0.0446243  0.0109625  -4.071 4.92e-05 ***
## P_Youth           0.0075522  0.0158173   0.477   0.6331
## P_Cook           -0.0439120  0.0109959  -3.993 6.81e-05 ***
## P_DIY            -0.0726329  0.0154558  -4.699 2.83e-06 ***
## P_Art             0.1339558  0.0128335  10.438  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.2027402)
##
##     Null deviance: 398.84  on 1599  degrees of freedom
## Residual deviance: 322.36  on 1590  degrees of freedom
## AIC: 1999.2
##
## Number of Fisher Scoring iterations: 2
```

```
print("Linear Synthetic balanced")
```

```
## [1] "Linear Synthetic balanced"
```

```
print(caret::confusionMatrix(predlm.rose,bbtest$Choice))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    0
##          1  158  638
##          0   46 1458
##
##                Accuracy : 0.7026
##                  95% CI : (0.6835, 0.7212)
##     No Information Rate : 0.9113
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2035
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7745
##             Specificity : 0.6956
##          Pos Pred Value : 0.1985
##          Neg Pred Value : 0.9694
##              Prevalence : 0.0887
##          Detection Rate : 0.0687
##    Detection Prevalence : 0.3461
##       Balanced Accuracy : 0.7351
##
##        'Positive' Class : 1
##
```

```
print("Logistic Synthetic balanced")
```

```
## [1] "Logistic Synthetic balanced"
```

```
print(caret::confusionMatrix(predlog.rose,bbtest$Choice))
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction    1    0
##          1  158  640
##          0   46 1456
##                                           
##                Accuracy : 0.7017          
##                  95% CI : (0.6826, 0.7204)
##     No Information Rate : 0.9113          
##     P-Value [Acc > NIR] : 1               
##                                           
##                   Kappa : 0.2027          
##                                           
##  Mcnemar's Test P-Value : <2e-16          
##                                           
##             Sensitivity : 0.7745          
##             Specificity : 0.6947          
##          Pos Pred Value : 0.1980          
##          Neg Pred Value : 0.9694          
##              Prevalence : 0.0887          
##          Detection Rate : 0.0687          
##    Detection Prevalence : 0.3470          
##       Balanced Accuracy : 0.7346          
##                                           
##        'Positive' Class : 1               
## 
```

```
print("SVM Synthetic balanced")
```

```
## [1] "SVM Synthetic balanced"
```

```
print(caret::confusionMatrix(predsvm.rose,bbtest$Choice))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    1    0
##          1  154  596
##          0   50 1500
##
##                Accuracy : 0.7191
##                  95% CI : (0.7003, 0.7374)
##     No Information Rate : 0.9113
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2131
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.75490
##             Specificity : 0.71565
##          Pos Pred Value : 0.20533
##          Neg Pred Value : 0.96774
##              Prevalence : 0.08870
##          Detection Rate : 0.06696
##    Detection Prevalence : 0.32609
##       Balanced Accuracy : 0.73528
##
##        'Positive' Class : 1
##
```

The variable with the greatest influence on the purchase of “The Art History of Florence” is unsurprisingly the number of Art books purchased by the customer (P_Art), followed in positive influence by the length of time since the customer last purchased a book (Last_purchase)

Interestingly, the attributes that made it less likely for a customer to buy the book were being male, or having purchased books other than art books.

In fact, a simple decision for determining the likelihood of buying an art book, BBBC could send a brochure to every female who has previously purchased an art book and achieve a positive prediction rate up to 40%

```
## [1] "Test observations that are Female with more than 0 Art Book purchases"
```

```
##
##   1   0
##  55 141
```

```
## [1] "Total test population"
```

```
##
##    1    0
##  204 2096
```

However, given that the company wants a generalized model for predicting which customers should be sent brochures to sell new books which may not necessarily always be art books, they should send the brochure to a sample of their customer base and then use the logistic model to analyze the purchase history of customers for that style of book, and then use that model to determine brochure recipients at the larger scale.

Appendix

Preprocessing the data

```
knitr::opts_chunk$set(echo = TRUE)
library(readxl)
library(caret)
library(Boruta)
library(Rcpp)
library(e1071)
library(ROSE)
setwd("~/GitHub/DA6813CaseStudies/Case2")
#setwd("~/MSDA/Fall 2021/GitHub")
#setwd("~/MSDA/Fall 2021/GitHub/DA6813CaseStudies")
#setwd("~/MSDA/Fall 2021/Data Analytics Applications/Case Study 1/DA6813CaseStudies/Case2")
```

```
set.seed(12345)
bbtrain<-read_excel('BBBC-Train.xlsx')
bbtest<-read_excel('BBBC-Test.xlsx')

# Check for Missing Values
#sum(is.na(bbtrain))
#sum(is.na(bbtest))


#remove the index column
bbtrain<-bbtrain[-1]
bbtest<-bbtest[-1]


bbtrain$Choice<-as.factor(bbtrain$Choice)
bbtest$Choice<-as.factor(bbtest$Choice)
```

```
#balance the target classes
bbtrain.over<-upSample(x=bbtrain[,2:ncol(bbtrain)],y=bbtrain$Choice)
bbtrain.over$Choice <- factor(bbtrain.over$Class)
bbtrain.over$Class<-NULL

bbtrain.rose=ROSE(Choice~.,data=bbtrain,seed=12345)$data
```

Training different models

```r
#Logistic Regression
# Train the unbalanced dataset for logistic regression model
log<-glm(Choice~.-First_purchase,data=bbtrain,family=binomial)
#summary(log)
# Train the Over-sampled for logistic regression model
log_over<-glm(Choice~.-First_purchase,data=bbtrain.over,family=binomial)
#summary(log_over)
# Train the Synthetically generated  dataset for logistic regression model
log_rose<-glm(Choice~.-First_purchase,data=bbtrain.rose,family=binomial)
#summary(log_rose)
```

```r
tunedsvm=tune("svm",Choice~.-First_purchase,data=bbtrain,kernel ="linear",ranges=list(cost=c( 0.001, 0.01, 1,5)))

svm<-tunedsvm$best.model

#summary(svm)

tunedsvm_over=tune("svm",Choice~.-First_purchase,data=bbtrain.over,kernel ="linear",ranges=list(cost=c( 0.001, 0.01, 1,5)))

svm_over<-tunedsvm_over$best.model

#summary(svm_over)

tunedsvm_rose=tune("svm",Choice~.-First_purchase,data=bbtrain.rose,kernel ="linear",ranges=list(cost=c( 0.001, 0.01, 1,5)))

svm_rose<-tunedsvm_rose$best.model
#summary(svm_rose)
```

```r
# Linear Regression
#convert factor back to numeric for linear regression
bbtrain$Choice<-as.numeric(as.character(bbtrain$Choice))
bbtrain.over$Choice<-as.numeric(as.character(bbtrain.over$Choice))
bbtrain.rose$Choice<-as.numeric(as.character(bbtrain.rose$Choice))

lm<-glm(Choice~.-First_purchase,data=bbtrain,family=gaussian)
lm_over<-glm(Choice~.-First_purchase,data=bbtrain.over,family=gaussian)
lm_rose<-glm(Choice~.-First_purchase,data=bbtrain.rose,family=gaussian)

#summary(lm)
#summary(lm_over)
#summary(lm_rose)
```

```
# Prediction using unbalanced Dataset
predlog<-predict(log,bbtest,type="response")
predsvm<-predict(svm,bbtest)
predlm<-predict(lm,bbtest)

predlog<-as.factor(ifelse(predlog>0.5,1,0))
predlm<-as.factor(ifelse(predlm>0.5,1,0))


# Prediction using Over sampled Dataset
predlog.over<-predict(log_over,bbtest,type="response")
predsvm.over<-predict(svm_over,bbtest)
predlm.over<-predict(lm_over,bbtest)

predlm.over<-as.factor(ifelse(predlm.over>0.5,1,0))
predlog.over<-as.factor(ifelse(predlog.over>0.5,1,0))

# Prediction using Synthetically Generated Dataset
predlog.rose<-predict(log_rose,bbtest,type="response")
predsvm.rose<-predict(svm_rose,bbtest)
predlm.rose<-predict(lm_rose,bbtest)


predlm.rose<-as.factor(ifelse(predlm.rose>0.5,1,0))
predlog.rose<-as.factor(ifelse(predlog.rose>0.5,1,0))
```