The University of Texas Rio Grande Valley

School of Mathematical and Statistical Science

Statistical Learning (MATH 6392)

Case Study 2

Name: Sergio Soto Quintero

UTRGV ID: 0375494

Summer II / August 2020

1. Introduction

Neuromarketing research shows physical advertisements have a pronounced effect on consumer decision-making.The most successful marketing campaigns are multifaceted endeavors, mixing various mediums to engage their target audience. Magazine ads, websites, billboards, direct mail, social media and smartphone apps are just a few ways companies can convert customers. To drive profits, businesses must allocate their marketing dollars across the right channels. Today, no brand can dispute the power of the digital world, but what about physical advertisements like direct mail and print ads? What gives them their edge? [1].

2. Literature Review

The U.S. Postal Service Office of Inspector General partnered with the Center for Neural Decision Making at Temple University's Fox School of Business in 2015 and again in 2019—to study the power of print and digital advertisements, and to dive deeper into how different generations react to them. Expanding on the original study, the 2019 research specifically analyzed the effect of print versus digital ads on young and old consumers. Research showed that physical advertisements were more effective in leaving a lasting impression than their digital counterparts, regardless of consumer age. [1]

Fig. 1 Direct mail is the singles largest category of US local advertising spend, at $38.5 billion in 2018. [2]

In the original study, physical advertisements were proven to have more influence than digital ads in a number of ways. Not only did participants spend more time with physical ads, they also remembered them more quickly and confidently. Physical ads also elicited a stronger emotional response than their digital counterparts and, overall, had a longer-lasting impact. Looking at brain activity, researchers discovered that participants showed a greater subconscious valuation and desire for products or services advertised in a physical format. [1]

This means physical ads are particularly effective in two stages of the consumer journey: exposure to information and retrieval of information. Digital ads trumped their physical counterparts in only one area: focused attention. Though participants did show more attention to digital ads, they gained the same amount of information from both types of advertisements.[1]

In 2018, the US Postal Service sent over 77 Billion pieces of marketing mail, a number that suggest a high volume, but if we consider that digital marketing channels have become increasingly crowded and expensive in most recent years, more marketers are turning their sights towards traditional channels or a combination of several marketing strategies (Fig.1). [2]

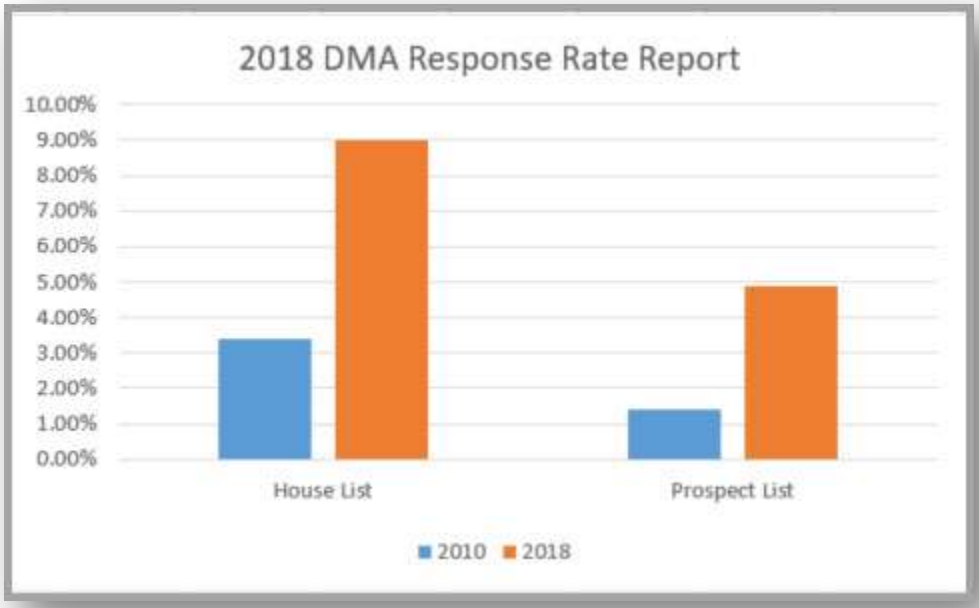


Fig. 2 The response rate of direct Direct mail was 4.9% for 2018 prospect lists, while the direct mail response rate for house lists was 9% (Numbers provided by USPS).

A recent study, published by the Canada Post Corporation in 2015 suggest that the best response rates to marketing strategies lie in the combination of two parameters: the intereaction provided by digital media and the action provided by the direct marketing. This neuromarketing study, provides evidence that direct mail influence the neurological processes that trigger action which is achieved in a greater scale compared to the digitial marketing. [3]

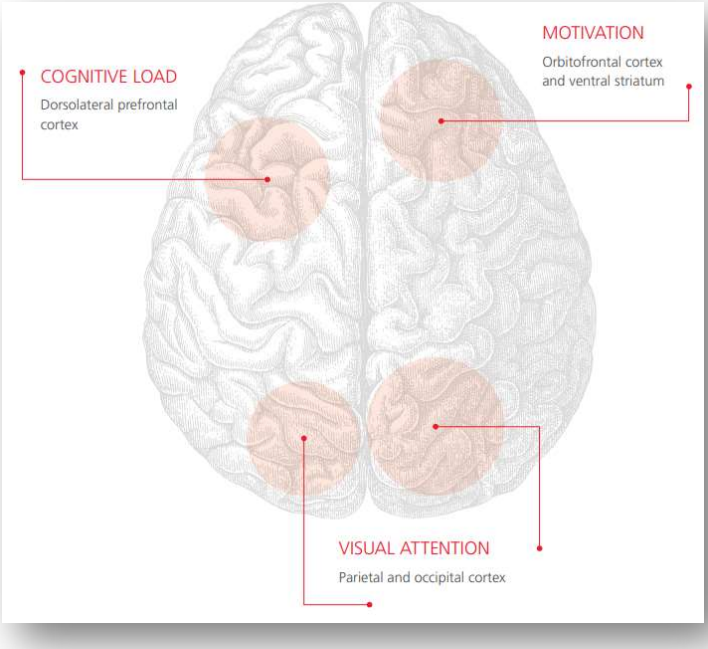


Fig. 3 Neuromarketing studies, focus on two key indicators of media effectiveness: ease of understanding and persuasiveness. They also looked at visual attention from the partipants.[3]

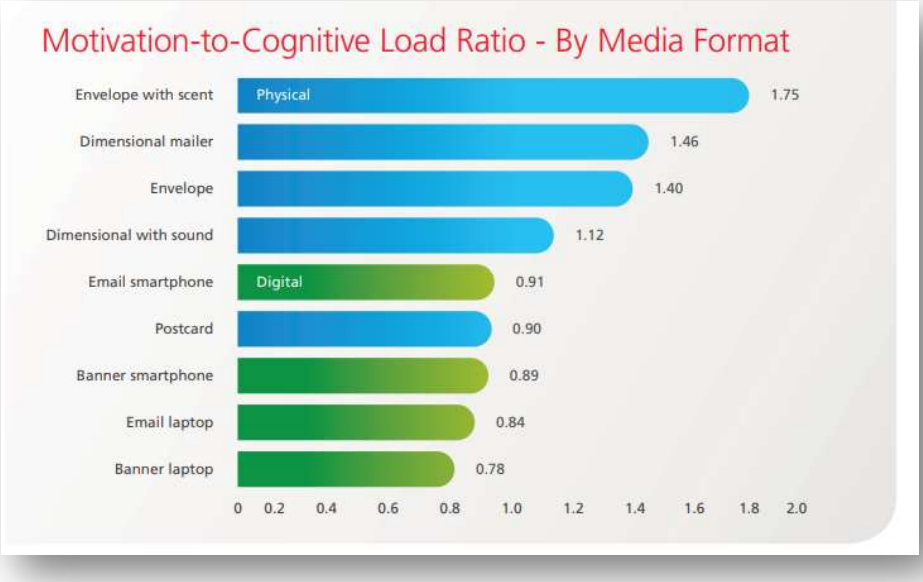


Fig. 4 Results from the motivation-to-cognitive load ratio by media format from the Canada Post Corporation study shows that envelope with scent is most effective at driving behavior, and in general, most physical media formats tested above the digital ones. [3]

3. Methods and Analysis

In this case study we will investigate the effectiveness of customer response to a direct mail marketing campaign using data sets from the insurance industry. The goal of the study is to improve the performance of future waves of this campaign by targeting people who are likely to take the offer.

A random forest and a support vector machine model will be used on a data set which has been splitted into a training set ("train.rda") and a validation set ("valid.rda") to estimate and compare the models. The training and validation set have 68 predictive variables and 10,000 observations each. The success of the model will be based on its ability to predict the probability that a costumer takes an offer, which will be captured by the **PURCHASE** predictor in the validation set.

Among the 68 predictive variables we encounter credit information, such as number of accounts, active account types, credit limits, credit utilization, age and location of the individual, etc. And because we have a large number of variables, we will first select the most significant variables and compare the same set of variables for each model.

We begin our analysis by removing all variables with an information value (IV) less than 0.05. The Information Value provides a great framework for exploratory analysis and variable screening for binary classifiers. It has been widely unded in the credit risk world for several decades. By removing the variables with IV lower than 0.05, we went from considering 68 predictors to 33.

Now, 33 predictors is still a large number of variables, so we proceed to shrink this number by eliminating highly correlated variables using variable clustering. The result gives a final list of 20 variables which will be used for the classification process.

"N_OPEN_REV_ACTS"	"TOT_HI_CRDT_CRDT_LMT"
"RATIO_BAL_TO_HI_CRDT"	"D_NA_M_SNC_MST_RCNT_ACT_OPN"
"AVG_BAL_ALL_PRM_BC_ACTS"	"M_SNC_MST_RCNT_ACT_OPN"
"M_SNC_OLDST_RETAIL_ACT_OPN"	"RATIO_RETAIL_BAL2HI_CRDT"
"PRCNT_OF_ACTS_NEVER_DLQNT"	"N_OF_SATISFY_FNC_REV_ACTS"
"M_SNC_MSTREC_INSTL_TRD_OPN"	"N_FNC_INSTLACTS"
"N_BC_ACTS_OPN_IN_24M"	"AVG_BAL_ALL_FNC_REV_ACTS"
"M_SNC_OLDST_MRTG_ACT_OPN"	"N_BANK_INSTLACTS"
"M_SNCOLDST_BNKINSTL_ACTOPN"	"D_REGION_A"
"PREM_BANKCARD_CRED_LMT"	"D_N_DISPUTED_ACTS"

Table 1. Final list of 20 variables that will be used for the Random Forest and SVM analysis.

After doing some cleaning of the data, including the renaming of the variable **PURCHASE** for **new_purchase_train** and **new_purchase_test** in the training and validation data sets respectively and changing some variables structure to factors we get to proceed with the Random Forest and SVM analysis.

Random Forest Analysis

The random forest is a classification **a**lgorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

We perform the Random Forest Analysis on the training set using variables the final 20 predictors (Table 1) and **new_purchase_train** as the response to the model. We use 1,001 trees instead of the suggested 10,001 to reduce the downtime associated with computer intensive calculations.

```
Call:
 randomForest(formula = new_purchase_train ~ ., data = mytraindata,      mtry = i, ntree = 1001, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 1001
No. of variables tried at each split: 13

        OOB estimate of  error rate: 18.29%
Confusion matrix:
     -1   1 class.error
-1 7610 367  0.04600727
1  1462 561  0.72268908
```

Table 2. Confusion matrix for the Random Forest for 13 different values of mtry on the training data set.

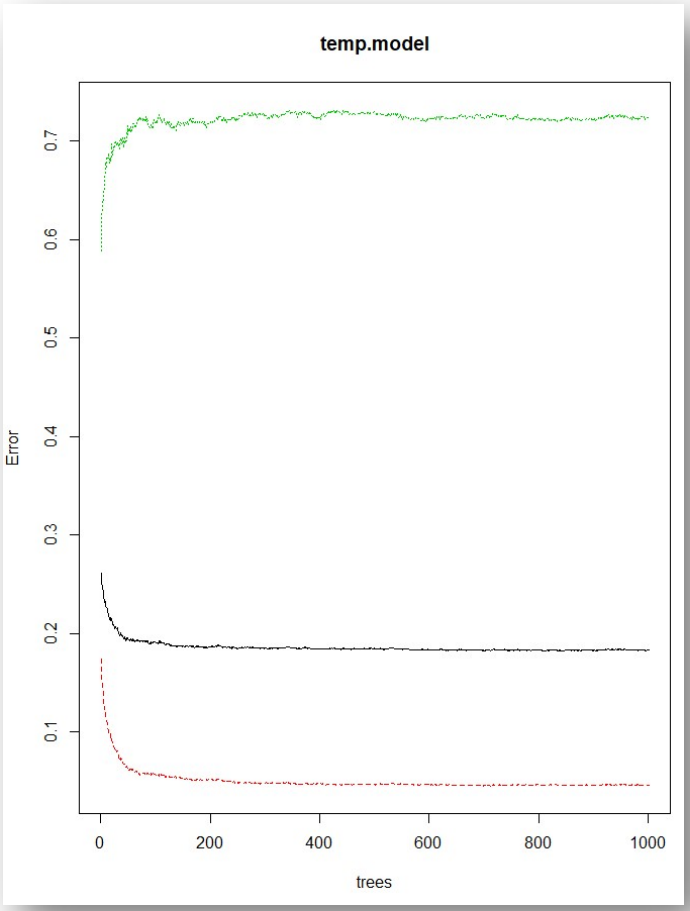


Fig. 5 Results from random forest for the train data set with 20 predictors, mtry = 10, and 1001 trees. The train error is displayed as a function of the number of trees. Each colored line corresponds to a different value of m, the number of predictors available for splitting at each interior tree node.

The Out-of-bag error is a method of measuring the prediction error of random forests, and we evaluate it for each mtry value (number of variables available for splitting at each tree node). The results are given below:

```
> oob.values
 [1] 0.2017 0.1840 0.1827 0.1809 0.1815 0.1818 0.1820 0.1814 0.1818 0.1800 0.1816 0.1835 0.1829
> #Find minimum error
> min(oob.values)
[1] 0.18
> # Find the optimal value for mtry
> which(oob.values == min(oob.values))
[1] 10
```

Table 3. Out-of-bag error (OOB) for the 13 Random Forest Models considered.

In Table 2, we observe that the minimum OOB value comes from model #10, so we use this model to create a variable importance plot to decide on the importance of the variables.

First, we run the Random Forest analysis using mtry = 10, ntree = 1001 on the training data.

```
Call:
 randomForest(formula = new_purchase_train ~ ., data = mytraindata,
               Type of random forest: classification
                     Number of trees: 1001
No. of variables tried at each split: 10

        OOB estimate of  error rate: 18.14%
Confusion matrix:
     -1   1 class.error
-1 7634 343  0.04299862
1  1471 552  0.72713791
```

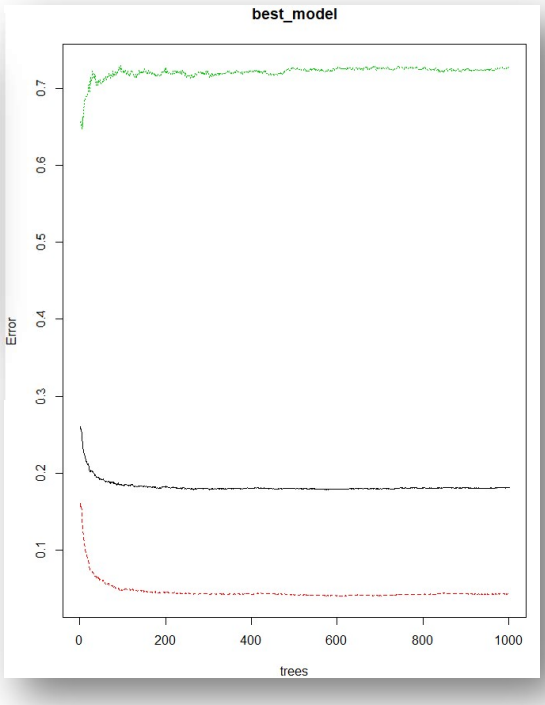


Fig. 6 Confusion matrix and plot of training error against number of trees using the model with the lowest OOB (mtry=10, ntree = 1001).

Fig. 7 Variable importance plot of the model with the lowest OOB after the Random Forest Analysis, The three most important variables to the model include: ***TOT_HI_CRDT_CRDT_LMT****,* ***N_OPEN_REV_ACTS****, and* ***M_SNC_OLDST_RETAIL_ACT_OPN***

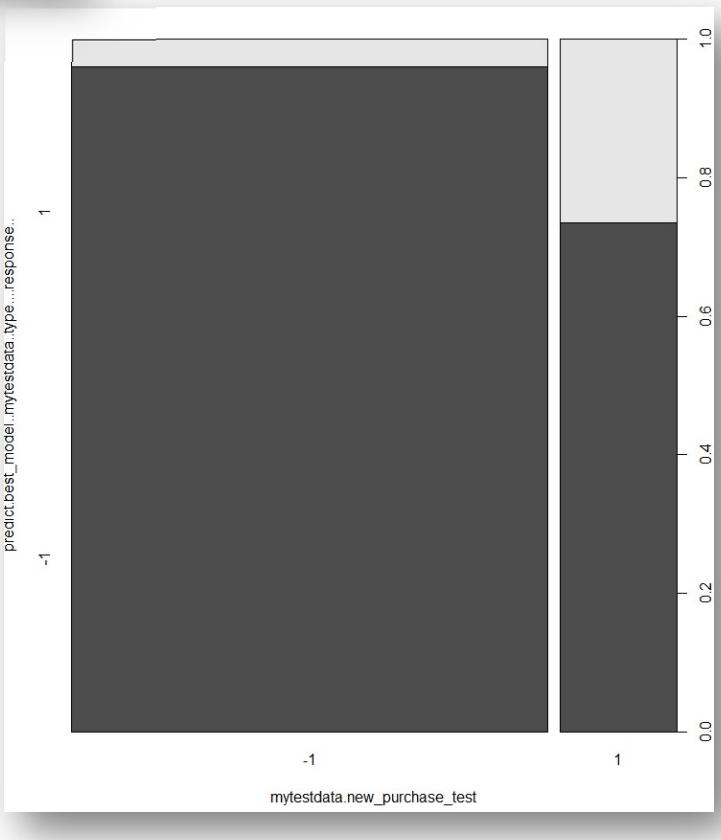
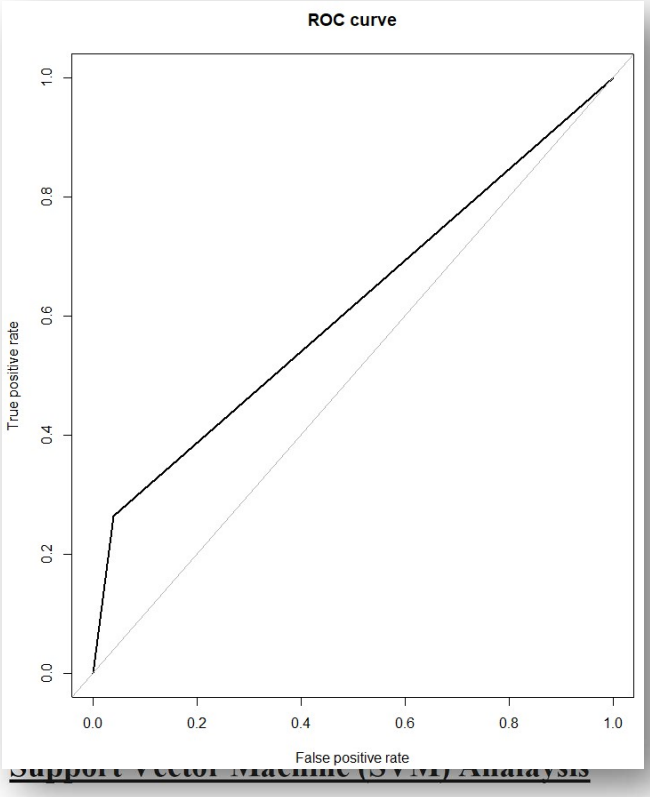


Fig. 8 Visualization of the correlation matrix of the model with the lowest OBB.

Now we use the “best model” to create a prediction using the valid data set.

```
Confusion Matrix and Statistics

                            predict.best_model..mytestdata..type....response..
mytestdata.new_purchase_test   -1    1
                           -1 7708  315
                           1  1454  523

               Accuracy : 0.8231
                 95% CI : (0.8155, 0.8305)
    No Information Rate : 0.9162
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2877

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8413
            Specificity : 0.6241
         Pos Pred Value : 0.9607
         Neg Pred Value : 0.2645
             Prevalence : 0.9162
         Detection Rate : 0.7708
   Detection Prevalence : 0.8023
      Balanced Accuracy : 0.7327

       'Positive' Class : -1
```

Table 4. Summary of the fitted model (prediction model) using the best model obtained from previous steps. The proportion of costumers that were predicted to take the offer out of those who actually took the offer (Sensitivity) equals 0.8413. The proportion of costumers that were predicted to not to take the offer out of those who actually did not take the offer (Specificity) equals 0.6341, and the percentage of all correct predictions that were made by the model (Accuracy) equals 0.8231.



$AUC_{randomforest}$ = 0.612

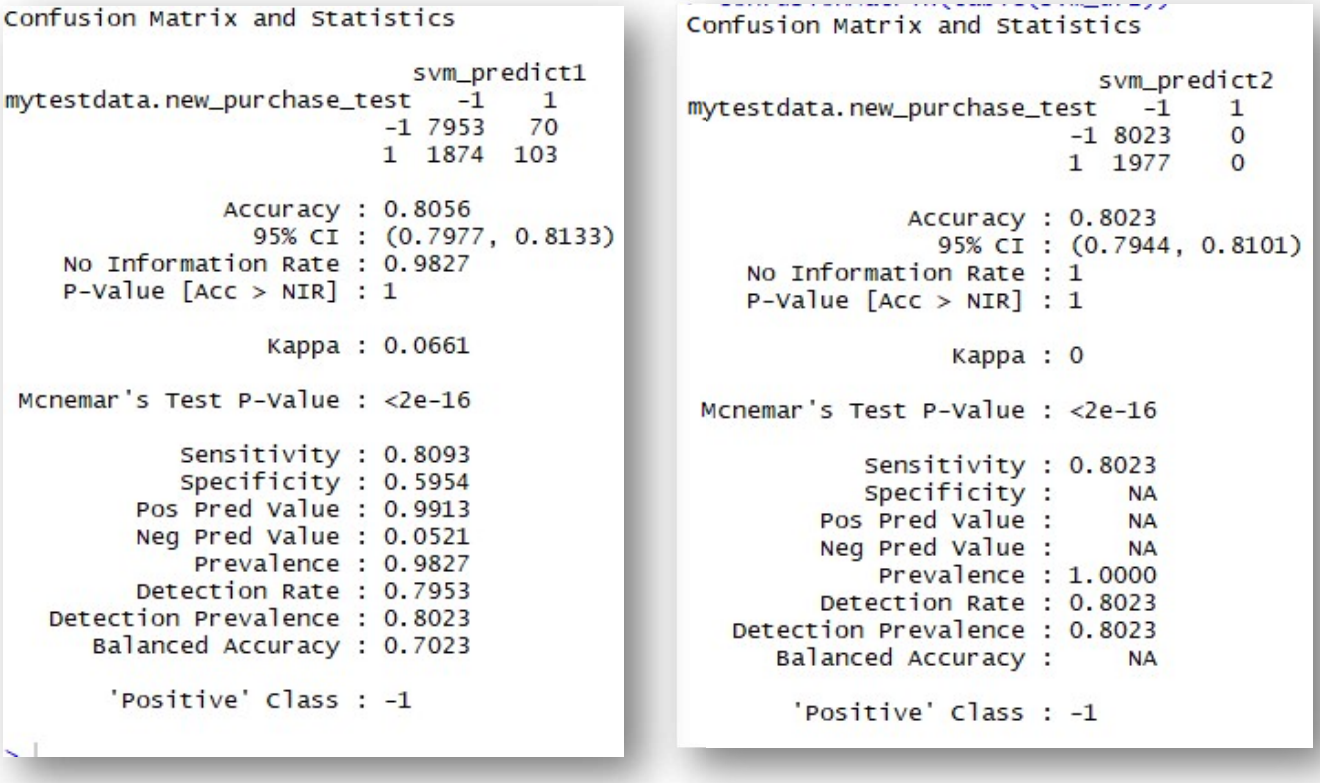
Fig 9. ROC curve of the best model. The are under the curve, the AUC value, is 0.612, which is a slightly accurate model.

Support Vector Machine (SVM) Analysis

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new data.

We will built two SVM classification models, one using a Polynomial kernel with degree 3 and the other one using a Gaussian Radial kernel with the training data, using the cost value of 0.01 on both models and a gamma value of 0.000001 for the second model.

The prediction for both models using the valid data set is presented in the summary tables from below:



```
Confusion Matrix and Statistics

                              svm_predict1
mytestdata.new_purchase_test   -1    1
                           -1 7953   70
                           1  1874  103

               Accuracy : 0.8056
                 95% CI : (0.7977, 0.8133)
    No Information Rate : 0.9827
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0661

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8093
            Specificity : 0.5954
         Pos Pred Value : 0.9913
         Neg Pred Value : 0.0521
             Prevalence : 0.9827
         Detection Rate : 0.7953
   Detection Prevalence : 0.8023
      Balanced Accuracy : 0.7023

       'Positive' Class : -1
```

```
Confusion Matrix and Statistics

                              svm_predict2
mytestdata.new_purchase_test   -1    1
                           -1 8023    0
                           1  1977    0

               Accuracy : 0.8023
                 95% CI : (0.7944, 0.8101)
    No Information Rate : 1
    P-Value [Acc > NIR] : 1

                  Kappa : 0

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8023
            Specificity :     NA
         Pos Pred Value :     NA
         Neg Pred Value :     NA
             Prevalence : 1.0000
         Detection Rate : 0.8023
   Detection Prevalence : 0.8023
      Balanced Accuracy :     NA

       'Positive' Class : -1
```

Table 5. Summary of the SVM prediction models. The list on the left side cotains the Polynomial kernel results, and the list on the right contains the Radial kernel results.

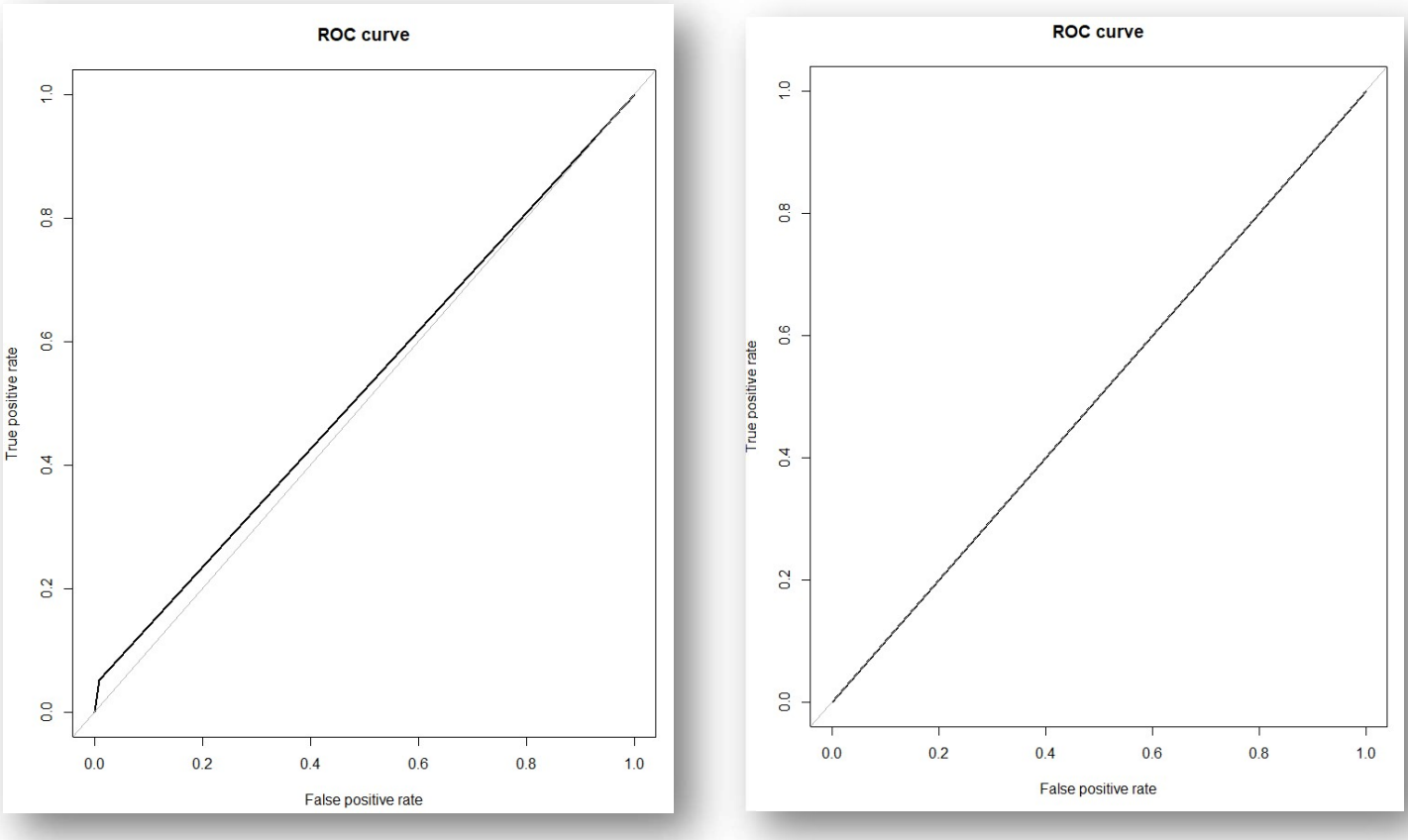


Table 5. ROC plot from the Radial SVM (Left) and the Polynomial SVM (Right). The closeness of the plot to the dotted-inclined line indicates a poor accuracy of the model.

The Area Under the ROC Curve (AUC), which gives is us the probability that the model ranks a random data point accurately for the two SVM models are:

$AUC_{polynomial} = 0.521$

$AUC_{radial} = 0.500$

4. Conclusion

On the basis of the findings, several conclusions concerning the interaction between the acceptance of an offers sent via direct mail and a list of 20 influential predictors The findings of this study suggest that the Random Forest Analysis results in a more accurate model than the SVM Analysis by comparing both model's AUC's. The results of the Random Forest Analysis indicates a sligh accuracy of predicting the accptence of an offer from direct mail giving an accuracy of 61%. The most influential variables to the model are: **TOT_HI_CRDT_CRDT_LMT**, **N_OPEN_REV_ACTS**, and **M_SNC_OLDST_RETAIL_ACT_OPN**. The study suggest than by focusing in the variables mentioned above a better accuracy in the predictions could be achieved.

5. Appendix

Appendix A. References

[1] USPS Office of Inspector General. (n.d.). *Is Direct Mail Advertising Effective? A Research Study.* Retrieved August 12, 2020, from https://www.uspsdelivers.com/why-direct-mail-is-more-memorable/

[2] William Boynton.(October 30th, 2019). *Important Statistics On Direct Mail.* Retrieved August 12, 2020, from https://www.postalytics.com/blog/statistics-on-direct-mail/

[3] Canada Post Corporation (2015). *A Bias for Action: The neuroscience behind the response-driving power of direct mail.* True Impact Marketing. Retrieved August 12th, 2020, from https://www.canadapost.ca/assets/pdf/blogs/CPC_Neuroscience_EN_150717.pdf

Appendix B. Code

```
# clear plots
if(!is.null(dev.list())) dev.off()

# clear console
cat("\014")

# clean workspace
rm(list=ls())

setwd("C:/Users/Checo/Desktop/Statistical Learning/Case Study 2")
getwd()

## 1. Read train and validation datasets
options(scipen=10)
train = readRDS(paste0("C:/Users/Checo/Desktop/Statistical Learning/Case Study
2/train.rda"))
valid = readRDS(paste0("C:/Users/Checo/Desktop/Statistical Learning/Case Study
2/valid.rda"))
dim(train)
dim(valid)

## 2. Information value (IV)
# Remove all variables with an IV less than 0.05 and create a new training and
validation data sets
# install.packages("Information")
library(Information)
IV = create_infotables(data=train, y ="PURCHASE", ncore=2)
#View(IV$Summary)

train_new = train[,c(subset(IV$Summary,IV>0.05)$Variable, "PURCHASE")]
dim(train_new)

valid_new=valid[,c(subset(IV$Summary, IV>0.05)$Variable, "PURCHASE")]
dim(valid_new)

## 3. Eliminate highly correlated variables using variable clustering
(ClustOfVar package)
# Select the most informative 20 variables to be used for the classification
using Variable Clustering

# install.packages("ClustOfVar")
# install.packages("reshape2")
# install.packages("plyr")
library(ClustOfVar)
library(reshape2)
library(plyr)

tree = hclustvar(train_new[,!(names(train_new) == "PURCHASE")])
nvars = 20
part_init = cutreevar(tree,nvars)$cluster
kmeans =
kmeansvar(X.quanti=train_new[,!(names(train_new)=="PURCHASE")],init=part_init)
clusters = cbind.data.frame(melt(kmeans$cluster),
row.names(melt(kmeans$cluster)))
names(clusters) = c("Cluster", "Variable")
clusters = join(clusters, IV$Summary, by="Variable", type="left")
clusters = clusters[order(clusters$Cluster),]
clusters$Rank = ave(-clusters$IV, clusters$Cluster, FUN=rank)
#View(clusters)
variables = as.character(subset(clusters, Rank==1)$Variable)
```

```
variables # Final 20 variables that will be used for classification purporses.

## 4.
# Create a new response variable called "NEWPurchase" using "PURCHASE" variable
in the train data set and add it to the train set
new_purchase_train = c(train_new$PURCHASE)
new_purchase_train = ifelse(train_new$PURCHASE == 1 , 1  , -1)
#new_purchase_train
cbind(train_new, new_purchase_train)
mytraindata= cbind(train_new[variables],new_purchase_train)
str(mytraindata)
# Change numeric values of predictors to factor
mytraindata$D_REGION_A = as.factor(mytraindata$D_REGION_A)
mytraindata$D_N_DISPUTED_ACTS = as.factor(mytraindata$D_N_DISPUTED_ACTS)
mytraindata$new_purchase_train = as.factor(mytraindata$new_purchase_train)

# Create a new response variable called "NEWPurchase" using "PURCHASE" variable
in the validation data set and add it to the validation set
new_purchase_test = c(valid_new$PURCHASE)
new_purchase_test = ifelse(valid_new$PURCHASE == 1 , 1  , -1)
# new_purchase_test
cbind(valid_new, new_purchase_test)
mytestdata= cbind(valid_new[variables],new_purchase_test)
# mytestdata
str(mytestdata)
# Change numeric predictors to factors
mytestdata$D_REGION_A = as.factor(mytestdata$D_REGION_A)
mytestdata$D_N_DISPUTED_ACTS = as.factor(mytestdata$D_N_DISPUTED_ACTS)
mytestdata$new_purchase_test = as.factor(mytestdata$new_purchase_test)

## 5.  Random Forest
# 5.1 Build up a Random forest using 1,001 trees. Use different "mtry" values
varying from 1 to 13. Evaluate the OOB error for each model
library(randomForest)
set.seed(123)
oob.values = vector(length=13)
for(i in 1:13) {
  temp.model = randomForest(new_purchase_train ~ ., data=mytraindata, mtry=i,
ntree=1001, importance = TRUE)
  oob.values[i] = temp.model$err.rate[nrow(temp.model$err.rate),1]
}
temp.model
plot(temp.model)
oob.values
#Find minimum error
min(oob.values)
# Find the optimal value for mtry
which(oob.values == min(oob.values))

# 5.2 Use model with lowest OBB error and create a variable importance plot
set.seed(123)
best_model = randomForest(new_purchase_train ~ . , data = mytraindata, mtry =
10, ntree = 1001)
best_model
plot(best_model)
importance(best_model)
order(importance(best_model))
varImpPlot(best_model)
varImpPlot(best_model,pch=18,col="red",cex=0.8)

# 5.3 Prediction
set.seed(123)
```

```
predict_model = data.frame(mytestdata$new_purchase_test,
predict(best_model,mytestdata,type="response"))
predict_model
plot(predict_model)

# 5.4 Evaluate the confusion matrix table and calculate the
sensitivity,specificity and accuracy
library(caret)
library(ggplot2)
library(lattice)
set.seed(123)
confusionMatrix(table(predict_model))

# 5.5 Create the ROC curve and evaluate the AUC value
# install.packages("ROSE")
library(ROSE)
set.seed(123)
roc = roc.curve(mytestdata$new_purchase_test ,
predict_model$predict.best_model..mytestdata..type....response..)
auc = roc$auc
auc

## 6 SVM classification models
#install.packages("e1071")
library(e1071)

# 6.1
# SVM - Polynomial Kernel
# R Code for training data set
set.seed(123)
svm.model1 <- svm(new_purchase_train ~., data=mytraindata, cost=0.01,
kernel="polynomial", degree=3, probability=TRUE)
# R Code for Prediction:
set.seed(123)
svm_predict1 = predict(svm.model1,newdata=mytestdata,probability=TRUE)

# SVM - Gausian radial Kernel
# R code for training data set
svm.model2 = svm(new_purchase_train ~. , data = mytraindata, cost = 0.01,
gamma=0.000001, kernel = "radial", probability = TRUE)
# R code for prediction
set.seed(123)
svm_predict2 = predict(svm.model2, newdata=mytestdata, probability = TRUE)

# 6.2 Evaluate the confuction table and calculate Sensitivity, Specificity and
Accuracy using the valid data set of prediction
set.seed(123)
svm_df1 = data.frame(mytestdata$new_purchase_test,svm_predict1)
svm_df2 = data.frame(mytestdata$new_purchase_test, svm_predict2)

confusionMatrix(table(svm_df1))
confusionMatrix(table(svm_df2))


# 6.3 Create the ROC curve and evaluate the AUC value
roc_svm1 = roc.curve( mytestdata$new_purchase_test,  svm_df1$svm_predict1 )
auc_svm1 = roc_svm1$auc
auc_svm1

roc_svm2 = roc.curve(mytestdata$new_purchase_test , svm_df2$svm_predict2 )
auc_svm2 = roc_svm2$auc
auc_svm2
```