Appendix 3 - Customer’s decision to open a term deposit account

## Part 4 - SVM model

### Step 1 : Collecting Data

Same as of Part 1

### Step 2 : Exploring, preprocessing and cleaning the data

Primary setup

knitr::opts\_knit$set(root.dir = normalizePath('D:\\Sobil'))  
remove(list = ls())  
set.seed(1)

loading all the libraries required

library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(kernlab)

##   
## Attaching package: 'kernlab'

## The following object is masked from 'package:ggplot2':  
##   
## alpha

rest same as part 1

source("bank\_import\_primaryExplore.R")

## 'data.frame': 41188 obs. of 21 variables:  
## $ age : int 56 57 37 40 56 45 59 41 24 25 ...  
## $ job : Factor w/ 12 levels "admin.","blue-collar",..: 4 8 8 1 8 8 1 2 10 8 ...  
## $ marital : Factor w/ 4 levels "divorced","married",..: 2 2 2 2 2 2 2 2 3 3 ...  
## $ education : Factor w/ 8 levels "basic.4y","basic.6y",..: 1 4 4 2 4 3 6 8 6 4 ...  
## $ default : Factor w/ 3 levels "no","unknown",..: 1 2 1 1 1 2 1 2 1 1 ...  
## $ housing : Factor w/ 3 levels "no","unknown",..: 1 1 3 1 1 1 1 1 3 3 ...  
## $ loan : Factor w/ 3 levels "no","unknown",..: 1 1 1 1 3 1 1 1 1 1 ...  
## $ contact : Factor w/ 2 levels "cellular","telephone": 2 2 2 2 2 2 2 2 2 2 ...  
## $ month : Factor w/ 10 levels "apr","aug","dec",..: 7 7 7 7 7 7 7 7 7 7 ...  
## $ day\_of\_week : Factor w/ 5 levels "fri","mon","thu",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ duration : int 261 149 226 151 307 198 139 217 380 50 ...  
## $ campaign : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ pdays : int 999 999 999 999 999 999 999 999 999 999 ...  
## $ previous : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ poutcome : Factor w/ 3 levels "failure","nonexistent",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ emp.var.rate : num 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 1.1 ...  
## $ cons.price.idx: num 94 94 94 94 94 ...  
## $ cons.conf.idx : num -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 -36.4 ...  
## $ euribor3m : num 4.86 4.86 4.86 4.86 4.86 ...  
## $ nr.employed : num 5191 5191 5191 5191 5191 ...  
## $ y : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...  
##   
##   
## Cell Contents  
## |-------------------------|  
## | N |  
## | N / Table Total |  
## |-------------------------|  
##   
##   
## Total Observations in Table: 41188   
##   
##   
## | no | yes |   
## |-----------|-----------|  
## | 36548 | 4640 |   
## | 0.887 | 0.113 |   
## |-----------|-----------|  
##   
##   
##   
##

### Step 3 - Data transformation & preparation

creating training and testing dataset from exisitng sample

set.seed(1)  
indx <- createDataPartition(bank$y, p = 0.8, list = FALSE)  
  
bank\_train <- bank[indx,]  
bank\_test <- bank[- indx,]  
  
prop.table(table(bank\_train$y))

##   
## Yes No   
## 0.1126521 0.8873479

prop.table(table(bank\_test$y))

##   
## Yes No   
## 0.1126624 0.8873376

### Step 4 : Training a model on the data

bank\_svm <- ksvm(y ~ ., data = bank\_train, kernel = "vanilladot")

## Setting default kernel parameters

bank\_svm

## Support Vector Machine object of class "ksvm"   
##   
## SV type: C-svc (classification)   
## parameter : cost C = 1   
##   
## Linear (vanilla) kernel function.   
##   
## Number of Support Vectors : 6674   
##   
## Objective Function Value : -6639.711   
## Training error : 0.096355

### Step 5 : Evaluating model performance

bank\_predict\_svm <- predict(object = bank\_svm, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_svm, bank\_test$y) # kappa = 0.4051, Sensitivity = 0.33297, Specificity = 0.98085

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 309 140  
## No 619 7169  
##   
## Accuracy : 0.9079   
## 95% CI : (0.9014, 0.914)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 7.895e-10   
##   
## Kappa : 0.4051   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.33297   
## Specificity : 0.98085   
## Pos Pred Value : 0.68820   
## Neg Pred Value : 0.92052   
## Prevalence : 0.11266   
## Detection Rate : 0.03751   
## Detection Prevalence : 0.05451   
## Balanced Accuracy : 0.65691   
##   
## 'Positive' Class : Yes   
##

### Step 6 : Improving model performance

#### 1) Method 1 : kernel = rbfdot

bank\_svm.rbf <- ksvm(y ~ . , data = bank\_train, kernel = "rbfdot")  
bank\_predict\_svm.rbf <- predict(object = bank\_svm.rbf, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_svm.rbf, bank\_test$y) # kappa = 0.4907, Sensitivity = 0.42996, Specificity = 0.97715

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 399 167  
## No 529 7142  
##   
## Accuracy : 0.9155   
## 95% CI : (0.9093, 0.9214)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.4907   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.42996   
## Specificity : 0.97715   
## Pos Pred Value : 0.70495   
## Neg Pred Value : 0.93104   
## Prevalence : 0.11266   
## Detection Rate : 0.04844   
## Detection Prevalence : 0.06871   
## Balanced Accuracy : 0.70355   
##   
## 'Positive' Class : Yes   
##

#### Method 2 : kernel = polydot

bank\_svm.poly <- ksvm(y ~ . , data = bank\_train, kernel = "polydot")

## Setting default kernel parameters

bank\_predict\_svm.poly <- predict(object = bank\_svm.poly, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_svm.poly, bank\_test$y) # kappa = 0.4051, Sensitivity = 0.33297, Specificity = 0.98085

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 309 140  
## No 619 7169  
##   
## Accuracy : 0.9079   
## 95% CI : (0.9014, 0.914)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 7.895e-10   
##   
## Kappa : 0.4051   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.33297   
## Specificity : 0.98085   
## Pos Pred Value : 0.68820   
## Neg Pred Value : 0.92052   
## Prevalence : 0.11266   
## Detection Rate : 0.03751   
## Detection Prevalence : 0.05451   
## Balanced Accuracy : 0.65691   
##   
## 'Positive' Class : Yes   
##

#### Method 3 : kernel = tanhdot

bank\_svm.tanh <- ksvm(y ~ . , data = bank\_train, kernel = "tanhdot")

## Setting default kernel parameters

bank\_predict\_svm.tanh <- predict(object = bank\_svm.tanh, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_svm.tanh, bank\_test$y) # kappa = 0.17, Sensitivity = 0.26078, Specificity = 0.90806

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 242 672  
## No 686 6637  
##   
## Accuracy : 0.8351   
## 95% CI : (0.8269, 0.8431)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 1.0000   
##   
## Kappa : 0.17   
##   
## Mcnemar's Test P-Value : 0.7243   
##   
## Sensitivity : 0.26078   
## Specificity : 0.90806   
## Pos Pred Value : 0.26477   
## Neg Pred Value : 0.90632   
## Prevalence : 0.11266   
## Detection Rate : 0.02938   
## Detection Prevalence : 0.11096   
## Balanced Accuracy : 0.58442   
##   
## 'Positive' Class : Yes   
##

from above 3 methods - trying adding cost param in kernel = rbfdot #### Method 4 : kernel = rbfdot & C = 10

bank\_svm.rbf10 <- ksvm(y ~ . , data = bank\_train, kernel = "rbfdot", C = 10)  
bank\_predict\_svm.rbf10 <- predict(object = bank\_svm.rbf10, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_svm.rbf10, bank\_test$y) # kappa = 0.4991, Sensitivity = 0.47737, Specificity = 0.96607

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 444 245  
## No 484 7064  
##   
## Accuracy : 0.9115   
## 95% CI : (0.9052, 0.9175)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 4.313e-13   
##   
## Kappa : 0.5013   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.47845   
## Specificity : 0.96648   
## Pos Pred Value : 0.64441   
## Neg Pred Value : 0.93588   
## Prevalence : 0.11266   
## Detection Rate : 0.05390   
## Detection Prevalence : 0.08365   
## Balanced Accuracy : 0.72246   
##   
## 'Positive' Class : Yes   
##

#### Method 5 : kernel = rbfdot & C = 50

bank\_svm.rbf50 <- ksvm(y ~ . , data = bank\_train, kernel = "rbfdot", C = 50)  
bank\_predict\_svm.rbf50 <- predict(object = bank\_svm.rbf50, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_svm.rbf50, bank\_test$y) # kappa = 0.4848, Sensitivity = 0.49461, Specificity = 0.95622

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 459 320  
## No 469 6989  
##   
## Accuracy : 0.9042   
## 95% CI : (0.8977, 0.9105)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 4.026e-07   
##   
## Kappa : 0.4848   
##   
## Mcnemar's Test P-Value : 1.372e-07   
##   
## Sensitivity : 0.49461   
## Specificity : 0.95622   
## Pos Pred Value : 0.58922   
## Neg Pred Value : 0.93711   
## Prevalence : 0.11266   
## Detection Rate : 0.05572   
## Detection Prevalence : 0.09457   
## Balanced Accuracy : 0.72542   
##   
## 'Positive' Class : Yes   
##

#### Method 6 : kernel = rbfdot & C = 100

bank\_svm.rbf100 <- ksvm(y ~ . , data = bank\_train, kernel = "rbfdot", C = 100)  
bank\_predict\_svm.rbf100 <- predict(object = bank\_svm.rbf100, newdata = bank\_test)  
caret::confusionMatrix(bank\_predict\_svm.rbf100, bank\_test$y) # kappa = 0.4853, Sensitivity = 0.51509, Specificity = 0.95034 ==> BEST

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 478 362  
## No 450 6947  
##   
## Accuracy : 0.9014   
## 95% CI : (0.8948, 0.9078)  
## No Information Rate : 0.8873   
## P-Value [Acc > NIR] : 2.088e-05   
##   
## Kappa : 0.4857   
##   
## Mcnemar's Test P-Value : 0.002265   
##   
## Sensitivity : 0.51509   
## Specificity : 0.95047   
## Pos Pred Value : 0.56905   
## Neg Pred Value : 0.93916   
## Prevalence : 0.11266   
## Detection Rate : 0.05803   
## Detection Prevalence : 0.10198   
## Balanced Accuracy : 0.73278   
##   
## 'Positive' Class : Yes   
##

confusionMatrix(bank\_predict\_svm.rbf100, bank\_test$y, mode="prec\_recall")

##Confusion Matrix and Statistics

##

## Reference

## Prediction Yes No

## Yes 478 362

## No 450 6947

##

## Accuracy : 0.9014

## 95% CI : (0.8948, 0.9078)

## No Information Rate : 0.8873

## P-Value [Acc > NIR] : 2.088e-05

## Kappa : 0.4857

##

## Mcnemar's Test P-Value : 0.002265

##

## Precision : 0.56905

## Recall : 0.51509

## F1 : 0.54072

## Prevalence : 0.11266

## Detection Rate : 0.05803

## Detection Prevalence : 0.10198

## Balanced Accuracy : 0.73278

##

## 'Positive' Class : Yes

##