Who are the best prospects for our term deposit marketing campaign?

Vanessa Okosun and Flore Zadji
MS in Business Analytics

TABLE OF CONTENTS

MERCERBANK & IT'S PRODUCTS

- MercerBank offers many services such as personalized financial services tailored to your needs.
- You can get concierge-like service and assistance in a wide array of financial matters ranging from deposit accounts, wire transfers and currency exchanges to investments, wealth planning and elder care.
- Also known as a fixed-term investment. It is the depositing of money into an account at a financial institution that offers a guaranteed return.

PROBLEM & OBJECTIVE

Problem:

- The bank has a product that is underperforming and not bringing in enough revenue.

Objective:

- To find the best candidate/prospect for the term deposit
- What is the best channel to reach out to the prospects?
- Is phone marketing the best channel for this marketing campaign?

PROJECT OVERVIEW

Target Right Prospects → **Term Deposit Yes/No Y(Target)**

Customer Info

Age
Job
Marital
Education
Default
Housing
Loan

Previous Campaign

Month
Dayofweek
Duration
Campaign
Pdays
Previous
Poutcome

Economic Index

Emp.var.rate
Cons.price.idx
Cons.conf.idx
Euribor3m
Nr.employed

OUR APPROACH

Analyzed dataset to select only variables that are important to our analysis and to detect anomalies in the variables.

Select type of model to be used based defined objective

Create an oversample data to counter bias towards ‘no’ in our dataset and run each model with both imbalance and balance dataset

Compare the performance of each model and Select the model that helps answer our objective.

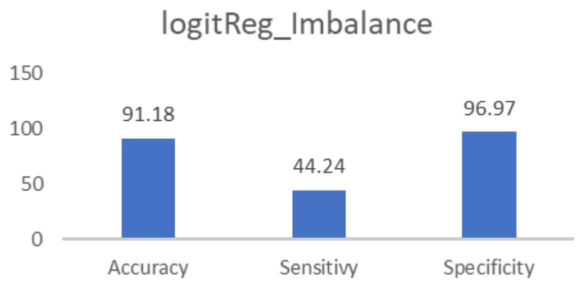
FINAL VARIABLES

- Y
- DEFAULT
- CONTACT
- DAY_OF_WEEK
- EDUCATION
- POUTCOME
- JOB
- MARTIAL
- LOAN
- HOUSING
- MONTH

Respondent No: 89%
Respondent Yes: 11%

MODEL PERFORMANCE

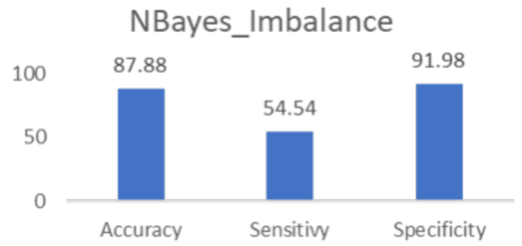
Model 1



Confusion Matrix and Statistics

Prediction	Reference 0	Reference 1
0	14276	1012
1	446	803

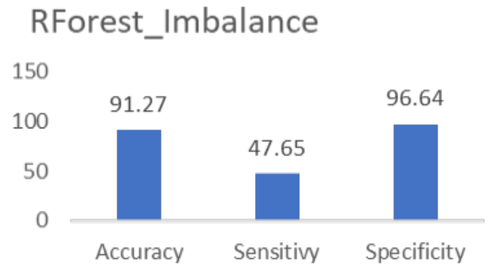
Model 2



Confusion Matrix and Statistics

Prediction	Reference 0	Reference 1
0	13542	825
1	1180	990

Model 3



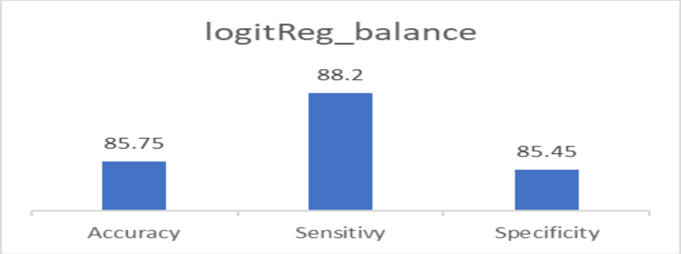
Confusion Matrix and Statistics

Prediction	Reference 0	Reference 1
0	14228	950
1	494	865

MODEL PERFORMANCE

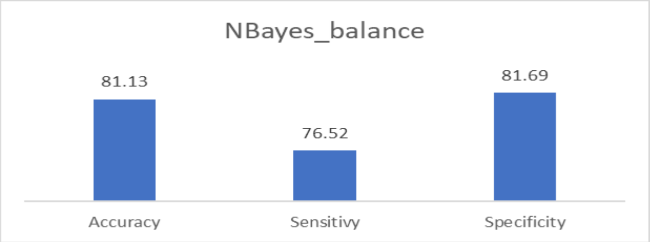
Respondent No: 51%
Respondent Yes: 49%

Model 1




```
          Reference
Prediction     0     1
         0 12580   214
         1  2142  1601
```

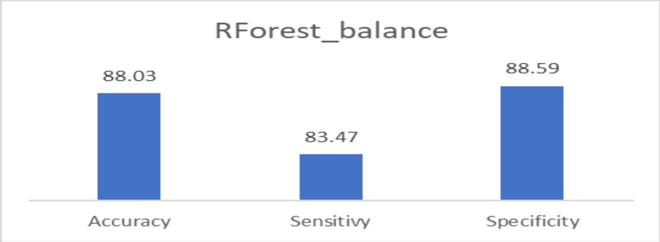
Model 2




```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 12027   426
         1  2695  1389
```

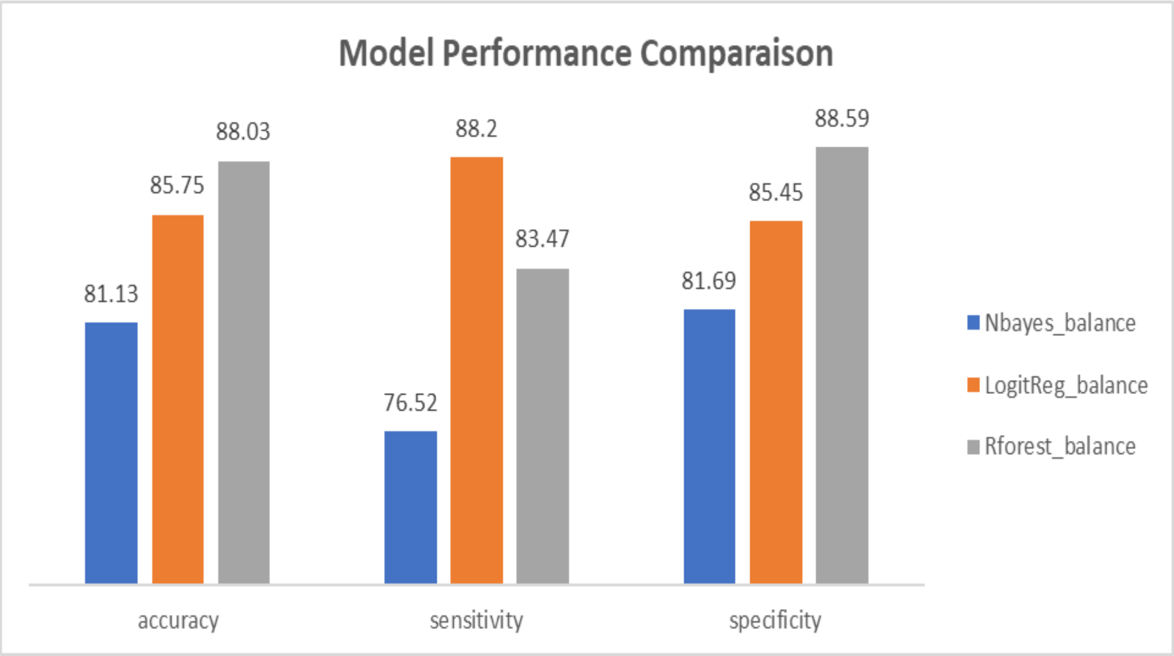
Model 3




```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 13043   300
         1  1679  1515
```

MODEL SELECTION



INSIGHTS

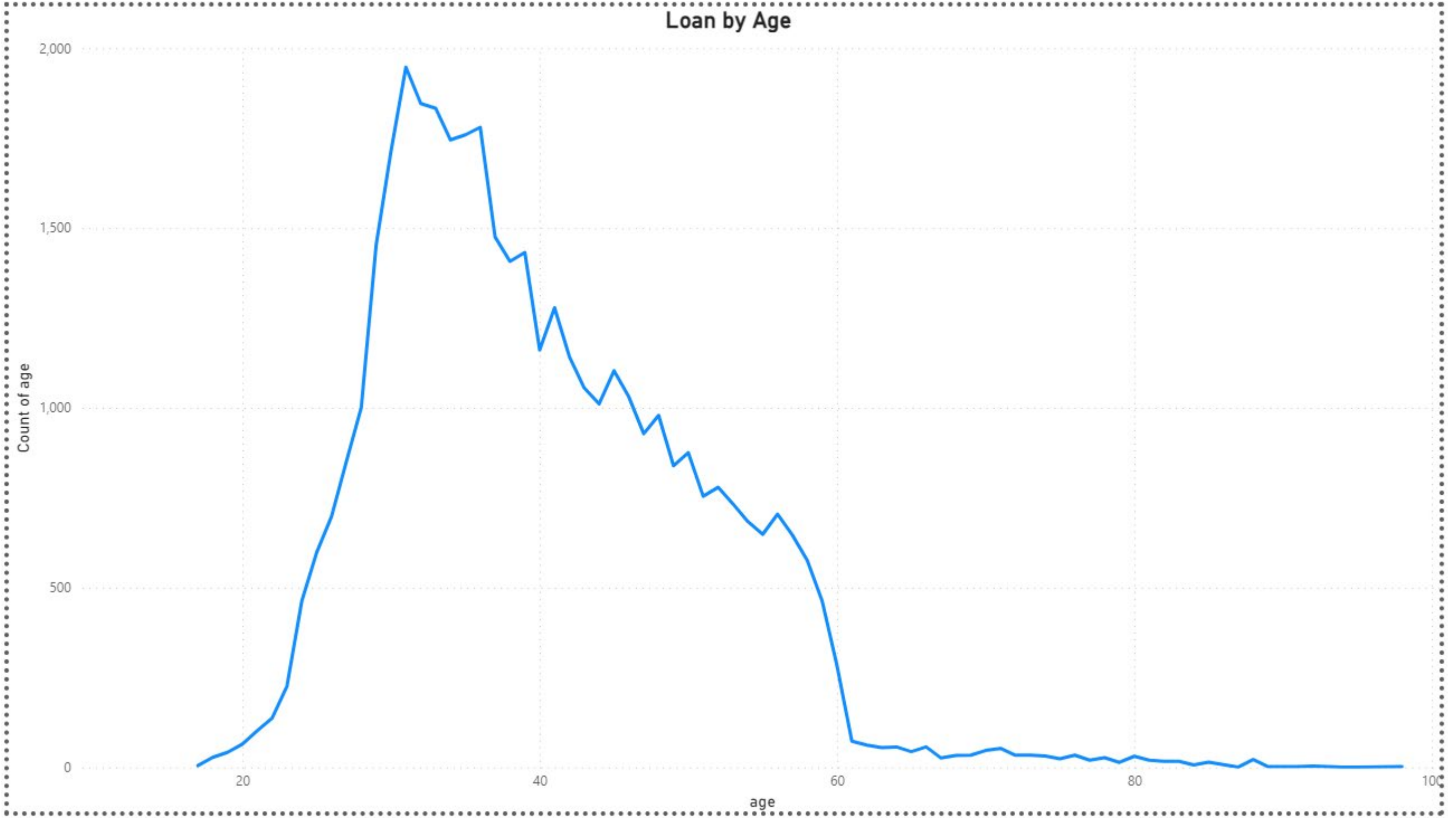
- Dominance of ‘no’ approved for term deposit created an imbalance in the dataset with bias towards ‘no’
- College degree retirees and students tend to favor term deposit
- Telephone outreach did not yield good result
- Monday and Friday were not good days to reach out to potential prospects
- Campaign launch in 2nd and 3rd quarter have better return than other quarters
- Campaign results decreased as the year ends out

RECOMMENDATION

- Retirees and college students are more inclined for this type of product. The business may need other communication channel other than landline to reach out to those prospects.
  - For students social medias channel or online campaign might be more effective in reaching out to that demography
  - For retirees Ads on Tv channels used by retirees, senior citizen clubs,
- Timing of the campaign matters. Study shows that campaign launched between second and third quarter yield better return than other quarters

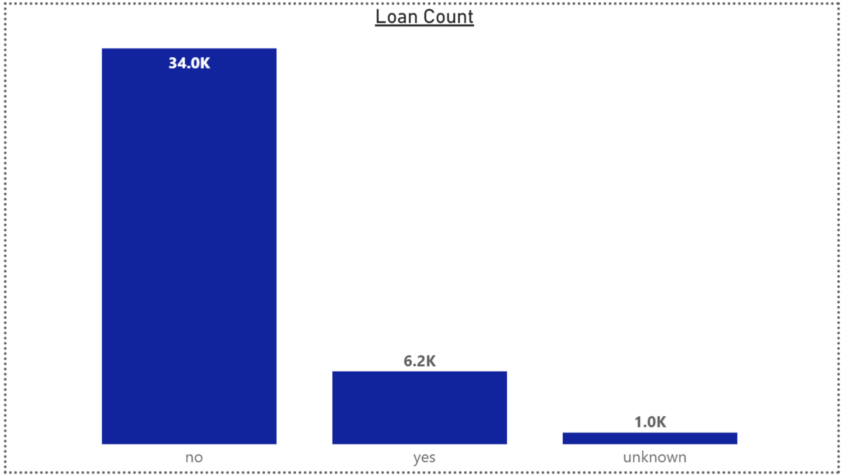
THANK YOU!

APPENDIX 1

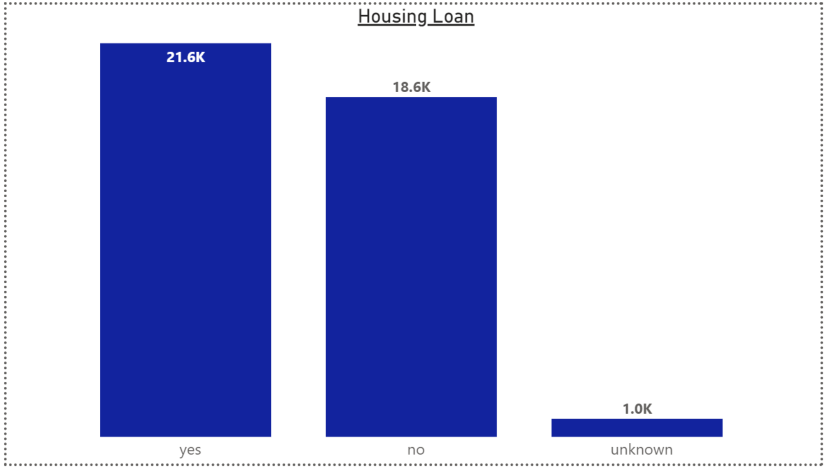


Most of the clients in this marketing campaign are between the ages mid 20s and early 60s.

APPENDIX 2

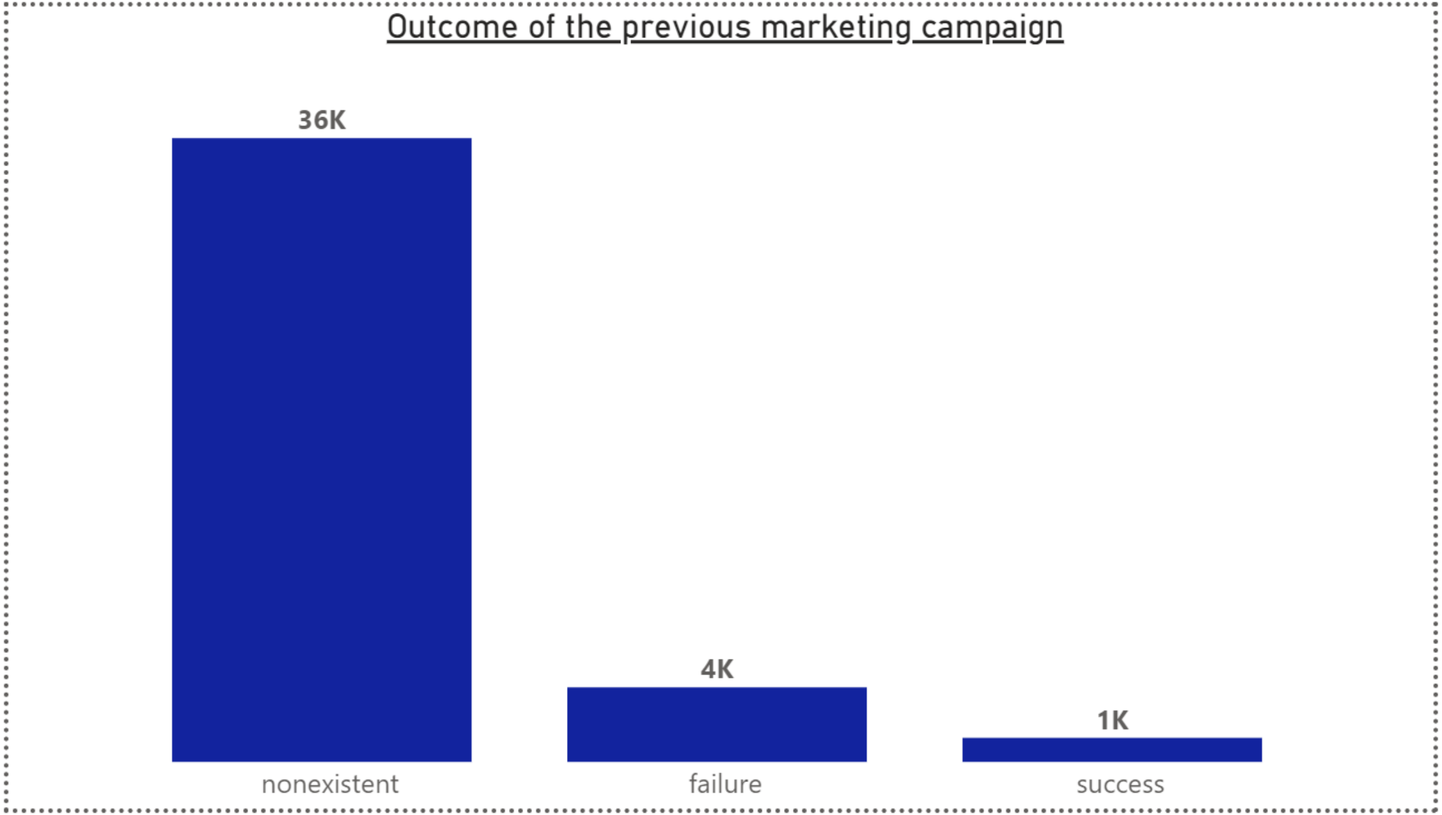


Out of the data of the clients, only 6.2K have a personal loan either though MercerBank or another insistution.



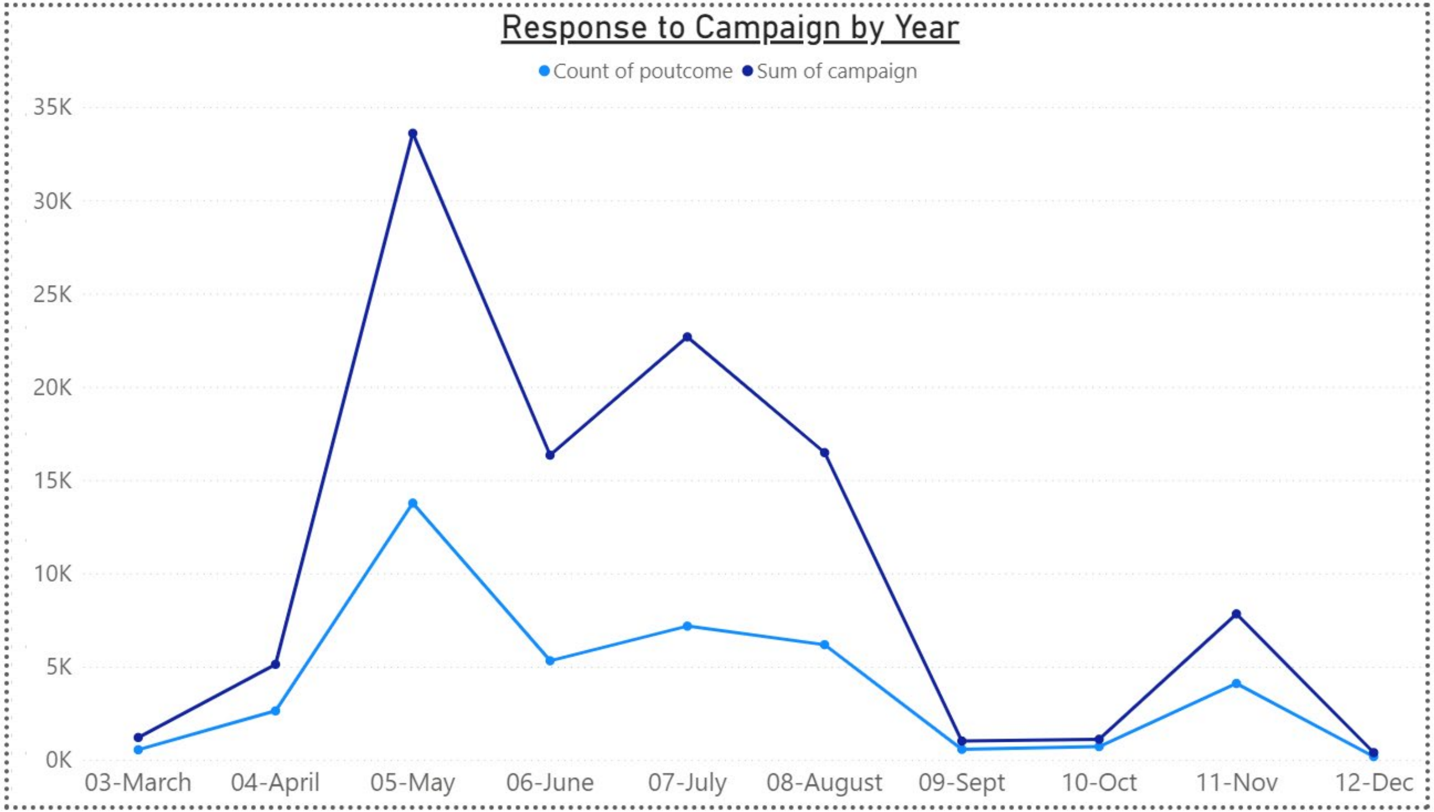
Out of the data of the clients, only 21.6K have a housing loan either though MercerBank or another insistution.

APPENDIX 3



Out of the data of the clients, only 6.2K have a personal loan either though MercerBank or another insistution.

APPENDIX 4

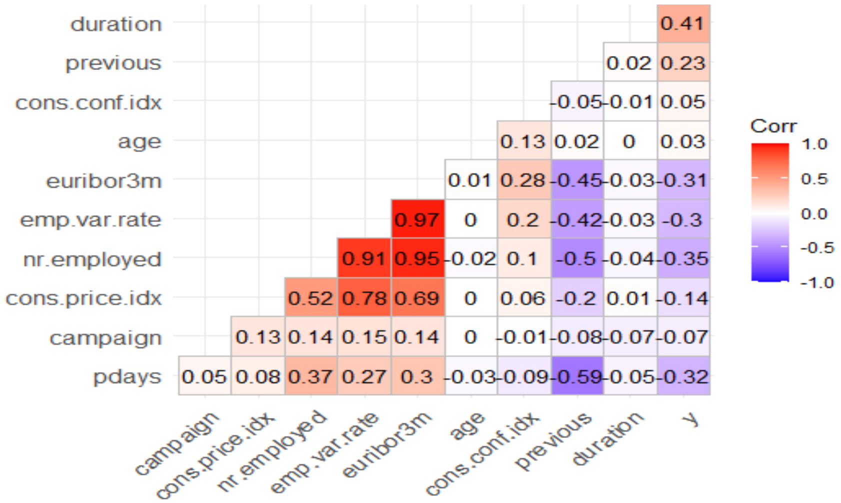


Response to campaign by year.

APPENDIX 5

```r
library(corrplot)
corrplot(cor(BMarketing.numer), method = "color")

library(ggcorrplot)
ggcorrplot(cor(BMarketing.numer),
           hc.order = TRUE,
          type = "lower",
          lab = TRUE)
```



Correlation analysis that show degree of correlation between variables

```
#### Dataset BMarketing.V1 exclude correlated variables
```{r}

to.drop <- c("pdays", "nr.employed", "age", "cons.conf.idx", "campaign", "euribor3m")

BMarketing.V1 <- BMarketing[ , -which(names(BMarketing) %in% to.drop)]

names(BMarketing.V1)

write.csv(BMarketing,'BMarketing.V1.csv')


str(BMarketing.V1)
```
```

APPENDIX 6

```
 [1] "job"              "marital"          "education"        "default"          "housing"
 [6] "loan"             "contact"          "month"            "day_of_week"      "duration"
[11] "previous"         "poutcome"         "emp.var.rate"     "cons.price.idx"   "y"
'data.frame':   41188 obs. of  15 variables:
 $ job           : chr  "housemaid" "services" "services" "admin." ...
 $ marital       : chr  "married" "married" "married" "married" ...
 $ education     : chr  "basic.4y" "high.school" "high.school" "basic.6y" ...
 $ default       : chr  "no" "unknown" "no" "no" ...
 $ housing       : chr  "no" "no" "yes" "no" ...
 $ loan          : chr  "no" "no" "no" "no" ...
 $ contact       : chr  "telephone" "telephone" "telephone" "telephone" ...
 $ month         : chr  "may" "may" "may" "may" ...
 $ day_of_week   : chr  "mon" "mon" "mon" "mon" ...
 $ duration      : int  261 149 226 151 307 198 139 217 380 50 ...
 $ previous      : int  0 0 0 0 0 0 0 0 0 0 ...
 $ poutcome      : chr  "nonexistent" "nonexistent" "nonexistent" "nonexistent" ...
 $ 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 ...
 $ y             : num  0 0 0 0 0 0 0 0 0 0 ...
```

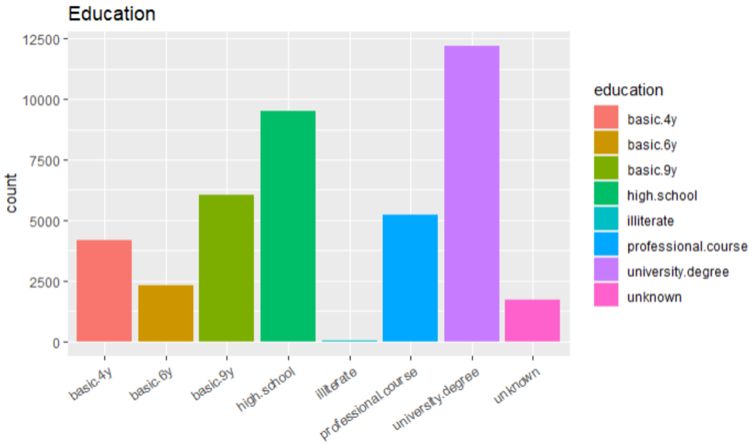
Reduce dimension of the dataset by removing insignificant variables

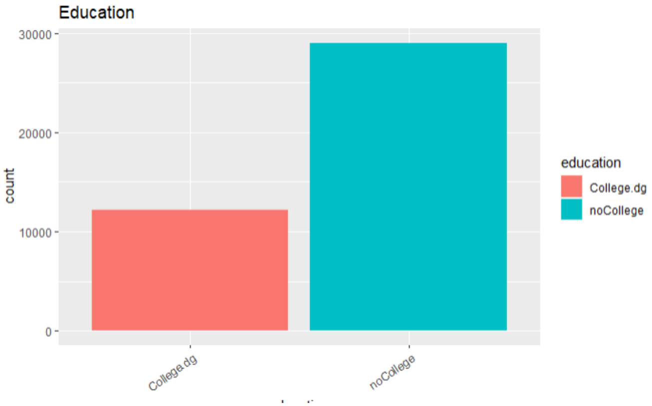
APPENDIX 7

```r
###recode job
bmarketing.v1$job <- case_when(
      bmarketing.v1$job == "unknown" ~ "unemployed",
      bmarketing.v1$job == "unemployed" ~ "unemployed",
      bmarketing.v1$job == "entrepreneur" ~ "entrepreneur",
      bmarketing.v1$job == "retired" ~ "retired",
      bmarketing.v1$job == "housemaid" ~ "housemaid",
      bmarketing.v1$job == "self-employed" ~ "self-employed",
      bmarketing.v1$job == "student" ~ "student",
      bmarketing.v1$job == "blue-collard" ~ "blue-collar",
      bmarketing.v1$job == "technician" ~ "blue-collar",
      bmarketing.v1$job == "services" ~ "blue-collar",
      TRUE ~ "white-collar")

 bmarketing.v1$job <- as.factor(bmarketing.v1$job)

 str(bmarketing.v1)
```





APPENDIX 8

```
```{r}
set.seed(456)

ind <- sample(2, nrow(bmarketing.v1), replace=TRUE, prob = c(0.6,0.4))

ov.train <- bmarketing.v1[ind==1, ]

ov.valid <- bmarketing.v1[ind==2,]
```

reponse prop in train
```{r}
prop.table(table(ov.train$y))

```
```

```
        0         1 
0.8874904 0.1125096 
```

Partition dataset into training and validation.
Dominance of ‘no’ 89% compared to 11% of ‘yes’

APPENDIX 9

```{r}
#install.packages("ROSE")
library(ROSE)

set.seed(455)

ind <- sample(2, nrow(bmarketing.v1), replace=TRUE, prob = c(0.6,0.4))

ov.train1 <- bmarketing.v1[ind==1, ]

ov.valid1 <- bmarketing.v1[ind==2,]


over <- ovun.sample(y~., data=ov.train1, method = "over", N=43652)$data
```

Created oversample dataset using ROSE library

APPENDIX 10

```
over.train1.glm <- glm(y~., data=over.1, family=binomial(link="logit"), na.action = na.exclude)

over.train1.glm.pred <- predict(over.train1.glm, ov.valid.1[, -14], type= "response")

over.train1.glm.pred2<-ifelse(over.train1.glm.pred > 0.5, 1, 0)

confusionMatrix(as.factor(over.train1.glm.pred2), as.factor(ov.valid.1$y), positive = '1')
```

```
Call:
glm(formula = y ~ ., family = binomial(link = "logit"), data = ov.train1,
    na.action = na.exclude)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-6.0662  -0.3073  -0.1839  -0.1333   3.1228

Coefficients:
                         Estimate    Std. Error z value             Pr(>|z|)
(Intercept)          -119.07731461    6.95187327 -17.129 < 0.0000000000000002 ***
jobentrepreneur        -0.05904870    0.15942571  -0.370             0.711097
jobhousemaid            0.03021855    0.17622789   0.171             0.863851
jobretired              0.28238801    0.11336849   2.491             0.012742 *
jobself-employed       -0.24567620    0.15091638  -1.628             0.103547
jobstudent              0.26572232    0.13755039   1.932             0.053382 .
jobunemployed           0.18345400    0.14468720   1.268             0.204821
jobwhite-collar        -0.10175141    0.06557159  -1.552             0.120720
maritalmarried          0.08453447    0.08844171   0.956             0.339163
maritalsingle           0.15587011    0.09575995   1.628             0.103585
educationnoCollege     -0.20119265    0.05841181  -3.444             0.000572 ***
defaultyes             -7.05867723  139.18406469  -0.051             0.959553
housingyes              0.01795312    0.05240211   0.343             0.731897
loanyes                -0.12702196    0.07411468  -1.714             0.086555 .
contacttelephone       -0.31445822    0.07927401  -3.967           0.00007287 ***
monthaug                1.00050924    0.11333054   8.828 < 0.0000000000000002 ***
monthdec                0.56068586    0.24793217   2.261             0.023731 *
```

Logistic regression results

APPENDIX 11

```
monthjul                0.51255341    0.11181368   4.584           0.00000456 ***
monthjun                0.11363681    0.11214853   1.013             0.310931    
monthmar                2.12806895    0.14643893  14.532 < 0.0000000000000002 ***
monthmay               -0.41789582    0.09532639  -4.384           0.00001166 ***
monthnov               -0.02925055    0.11739419  -0.249             0.803233    
monthoct                0.65145511    0.14055396   4.635           0.00000357 ***
monthsep                0.40410597    0.15287651   2.643             0.008209 ** 
day_of_weekmon         -0.17024515    0.08424612  -2.021             0.043300 *  
day_of_weekthu          0.01479917    0.08182799   0.181             0.856480    
day_of_weektue          0.03357382    0.08426022   0.398             0.690296    
day_of_weekwed          0.12157486    0.08394268   1.448             0.147531    
duration                0.00477753    0.00009713  49.187 < 0.0000000000000002 ***
previous                0.04978504    0.07360005   0.676             0.498770    
poutcomenonexistent     0.54290571    0.12181774   4.457           0.00000832 ***
poutcomesuccess         1.84553501    0.11024795  16.740 < 0.0000000000000002 ***
emp.var.rate           -1.02954020    0.03147529 -32.709 < 0.0000000000000002 ***
cons.price.idx          1.22184691    0.07427177  16.451 < 0.0000000000000002 ***
---
```

Logistic regression results

APPENDIX 12

```
confusionMatrix(as.factor(over.train1.glm.pred2), as.factor(ov.valid.1$y), positive = '1')
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurredConfusion Matrix and Statistics

          Reference
Prediction     0     1
         0 12580   214
         1  2142  1601

               Accuracy : 0.8575
                 95% CI : (0.8521, 0.8628)
    No Information Rate : 0.8902
    P-Value [Acc > NIR] : 1

                  Kappa : 0.5026

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.88209
            Specificity : 0.85450
         Pos Pred Value : 0.42773
         Neg Pred Value : 0.98327
             Prevalence : 0.10975
         Detection Rate : 0.09681
   Detection Prevalence : 0.22634
      Balanced Accuracy : 0.86830

       'Positive' Class : 1
```

Logistic regression confusion matrix for balance dataset

APPENDIX 13

```r
ov.train.nbayes <- naiveBayes(y~., data= nb.over)

ov.nb.pred <- predict(ov.train.nbayes, newdata= nb.valid1, type= "class")

confusionMatrix(ov.nb.pred, nb.valid1$y, positive ='1')
```

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 12027   426
         1  2695  1389

               Accuracy : 0.8113
                 95% CI : (0.8052, 0.8172)
    No Information Rate : 0.8902
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3761

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.76529
            Specificity : 0.81694
         Pos Pred Value : 0.34011
         Neg Pred Value : 0.96579
             Prevalence : 0.10975
         Detection Rate : 0.08399
   Detection Prevalence : 0.24696
      Balanced Accuracy : 0.79111

       'Positive' Class : 1
```

Naive Bayes confusion matrix with balance dataset

APPENDIX 14

```r
## confusion matrix with balance dataset

rfover <- randomForest(y~., data=over)

confusionMatrix(predict(rfover, ov.valid1), ov.valid1$y, positive='1')
```

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 13043   300
         1  1679  1515

               Accuracy : 0.8803
                 95% CI : (0.8753, 0.8852)
    No Information Rate : 0.8902
    P-Value [Acc > NIR] : 1

                  Kappa : 0.5406

 Mcnemar's Test P-Value : <0.0000000000000002

            Sensitivity : 0.83471
            Specificity : 0.88595
         Pos Pred Value : 0.47433
         Neg Pred Value : 0.97752
             Prevalence : 0.10975
         Detection Rate : 0.09161
   Detection Prevalence : 0.19314
      Balanced Accuracy : 0.86033

       'Positive' Class : 1
```

APPENDIX 15

```{r}
#install.packages("skimr")
library(skimr)

skim(BMarketing)

#### Stats for categorical variables
```

	skim_variable <chr>	n_missing <int>	complete_rate <dbl>	mean <dbl>	sd <dbl>	p0 <dbl>	p25 <dbl>
1	age	0	1	40.0240604	10.4212500	17.000	32.000
2	duration	0	1	258.2850102	259.2792488	0.000	102.000
3	campaign	0	1	2.5675925	2.7700135	1.000	1.000
4	pdays	0	1	962.4754540	186.9109073	0.000	999.000
5	previous	0	1	0.1729630	0.4949011	0.000	0.000
6	emp.var.rate	0	1	0.0818855	1.5709597	-3.400	-1.800
7	cons.price.idx	0	1	93.5756644	0.5788400	92.201	93.075
8	cons.conf.idx	0	1	-40.5026003	4.6281979	-50.800	-42.700
9	euribor3m	0	1	3.6212908	1.7344474	0.634	1.344
10	nr.employed	0	1	5167.0359109	72.2515277	4963.600	5099.100

1-10 of 11 rows | 1-8 of 11 columns

Previous 1 2 Next

```
names(BMarketing)
str(BMarketing)
#dim(BMarketing)
BMarketing.na <- na.omit(BMarketing)
#dim(BMarketing.na)
```

```
 [1] "age"            "job"             "marital"        "education"      "default"
 [6] "housing"        "loan"            "contact"        "month"          "day_of_week"
[11] "duration"       "campaign"        "pdays"          "previous"       "poutcome"
[16] "emp.var.rate"   "cons.price.idx" "cons.conf.idx" "euribor3m"      "nr.employed"
[21] "y"
'data.frame':   41188 obs. of  21 variables:
 $ age           : int  56 57 37 40 56 45 59 41 24 25 ...
 $ job           : chr  "housemaid" "services" "services" "admin." ...
 $ marital       : chr  "married" "married" "married" "married" ...
 $ education     : chr  "basic.4y" "high.school" "high.school" "basic.6y" ...
 $ default       : chr  "no" "unknown" "no" "no" ...
 $ housing       : chr  "no" "no" "yes" "no" ...
 $ loan          : chr  "no" "no" "no" "no" ...
 $ contact       : chr  "telephone" "telephone" "telephone" "telephone" ...
 $ month         : chr  "may" "may" "may" "may" ...
 $ day_of_week   : chr  "mon" "mon" "mon" "mon" ...
 $ 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      : chr  "nonexistent" "nonexistent" "nonexistent" "nonexistent" ...
 $ 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             : chr  "no" "no" "no" "no" ...
```

APPENDIX 16