PROJECT ON TELE-MARKETING CAMPAIGNS OF EUROPEAN BANKING INSTITUTIONS

BY

NIKHIL AIVALLI

Problem statement:

The data is about telemarketing campaigns of a European banking institution. The European bank wants to predict which clients will secure a term deposit based on a set of information on client and purchase of term deposit. The marketing is usually based on phone calls. Often, a client need to be persuaded multiple times in order to assess if the product (bank term deposit) would be or not subscribed. Predictive modelling approach will help the bank to manage their telemarketing campaign efficiently.

Read data

```
project<-read.csv('D:/IMARTICUS/final project/bank.csv',na.strings = c(""," ","NA"))
```

viewing the dataset

```
View(project)
```

Imputation

```
# checking for the missing values
colSums(is.na(project))

##            age            job        marital      education        default
##              0            330             80           1731           8597
##        housing           loan        contact          month    day_of_week
##            990            990              0              0              0
##       duration       campaign          pdays       previous       poutcome
##              0              0              0              0              0
##   emp.var.rate cons.price.idx  cons.conf.idx      euribor3m    nr.employed
##              0              0              0              0              0
##              y
##              0

#checking the diminsions of dataset
dim(project)

## [1] 41188    21
```

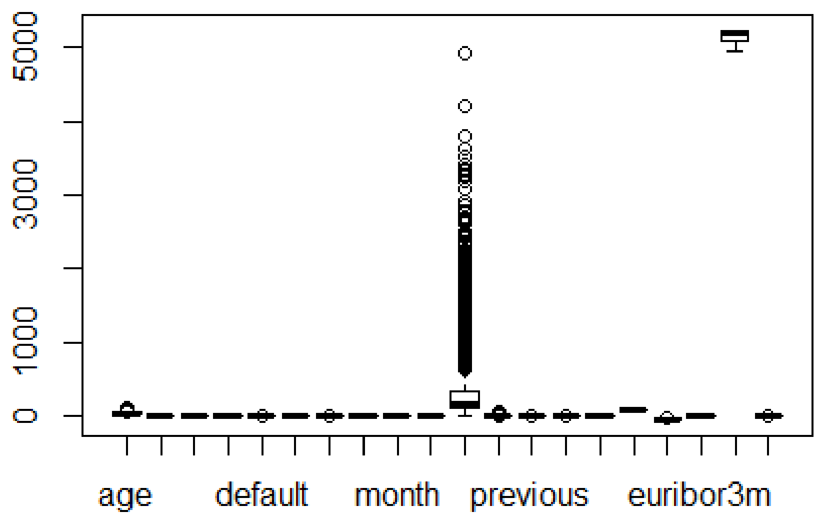
```r
# removing the unique columns
project$pdays<-NULL

# mode function

mode <- function(v){
  uniqv<-unique(v)
  uniqv[which.max(tabulate(match(v,uniqv)))]
}

project$job[is.na(project$job)]<-mode(project$job)
project$marital[is.na(project$marital)]<-mode(project$marital)
project$education[is.na(project$education)]<-mode(project$education)
project$default[is.na(project$default)]<-mode(project$default)
project$housing[is.na(project$housing)]<-mode(project$housing)
project$loan[is.na(project$loan)]<-mode(project$loan)

#chwcking for NA's in main dataset
colSums(is.na(project))
```

```
##            age            job        marital      education        default
##              0              0              0              0              0
##        housing           loan        contact          month    day_of_week
##              0              0              0              0              0
##       duration       campaign       previous       poutcome   emp.var.rate
##              0              0              0              0              0
## cons.price.idx  cons.conf.idx      euribor3m    nr.employed              y
##              0              0              0              0              0
```

```r
# creating the levels

levels(project$job)<-1:11
levels(project$marital)<-1:3
levels(project$education)<-1:7
levels(project$default)<-1:2
levels(project$housing)<-1:2
levels(project$loan)<-1:2
levels(project$contact)<-1:2
levels(project$month)<-1:10
levels(project$day_of_week)<-1:5
levels(project$poutcome)<-1:3
levels(project$y)<-0:1

# checking the class of the dependent variable
class(project$y)
```

```
## [1] "factor"
```

```r
# checking for the outliers
boxplot(project)
```




```
# scaling the dataset

project$age<-scale(project$age)
project$duration<-scale(project$duration)
project$cons.price.idx<-scale(project$cons.price.idx)
project$cons.conf.idx<-scale(project$cons.conf.idx)
project$euribor3m<-scale(project$euribor3m)
project$nr.employed<-scale(project$nr.employed)
```

splitting of the dataset into train and validate set

```
library(caTools)

## Warning: package 'caTools' was built under R version 3.4.4

set.seed(100)
split<- sample.split(project$y , SplitRatio = 0.70)
train_set<-subset(project,split ==TRUE)
val_set<-subset(project,split ==FALSE)
#View(val_set)
```

Building the Models on train set

```
attach(train_set)
```

1 Classification using Logistic regression model

```
model <-glm(y~ . , family = binomial(link = 'logit'), data = train_set)
summary(model)  #aic = 11959

##
## Call:
## glm(formula = y ~ ., family = binomial(link = "logit"), data = train_set)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.9927  -0.2962  -0.1830  -0.1318   3.1117
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.126e+00  2.033e-01 -15.380  < 2e-16 ***
## age            -3.053e-02  2.999e-02  -1.018  0.30874
## job2           -2.546e-01  9.293e-02  -2.740  0.00614 **
## job3           -2.658e-01  1.550e-01  -1.715  0.08636 .
## job4            5.550e-02  1.751e-01   0.317  0.75119
## job5            2.009e-02  1.014e-01   0.198  0.84302
## job6            2.330e-01  1.268e-01   1.838  0.06609 .
## job7           -1.836e-01  1.433e-01  -1.281  0.20028
## job8           -1.527e-01  1.035e-01  -1.476  0.13994
## job9            2.833e-04  1.311e-01   0.002  0.99828
## job10          -1.293e-02  8.488e-02  -0.152  0.87896
## job11           1.356e-01  1.509e-01   0.898  0.36900
## marital2       -7.225e-02  8.230e-02  -0.878  0.38000
## marital3        4.120e-02  9.378e-02   0.439  0.66039
## education2      3.542e-02  1.425e-01   0.249  0.80369
## education3     -3.182e-02  1.131e-01  -0.281  0.77840
## education4     -5.013e-03  1.089e-01  -0.046  0.96328
## education5      1.104e+00  8.573e-01   1.287  0.19792
## education6      4.409e-02  1.206e-01   0.365  0.71477
## education7      1.898e-01  1.056e-01   1.798  0.07217 .
## default2       -7.245e+00  1.390e+02  -0.052  0.95844
## housing2       -6.066e-03  4.917e-02  -0.123  0.90182
## loan2          -1.417e-01  7.096e-02  -1.997  0.04578 *
## contact2       -6.615e-01  9.201e-02  -7.189 6.53e-13 ***
## month2          9.214e-01  1.442e-01   6.389 1.67e-10 ***
## month3          4.666e-01  2.521e-01   1.851  0.06424 .
## month4          1.298e-01  1.158e-01   1.121  0.26233
## month5         -4.199e-01  1.508e-01  -2.785  0.00535 **
## month6          2.061e+00  1.728e-01  11.930  < 2e-16 ***
## month7         -4.382e-01  9.941e-02  -4.408 1.04e-05 ***
## month8         -4.168e-01  1.457e-01  -2.860  0.00423 **
## month9          2.519e-01  1.838e-01   1.371  0.17047
## month10         4.346e-01  2.126e-01   2.044  0.04095 *
## day_of_week2   -2.139e-01  8.056e-02  -2.656  0.00791 **
## day_of_week3    7.549e-02  7.719e-02   0.978  0.32813
## day_of_week4    7.937e-02  7.950e-02   0.998  0.31812
```

```
## day_of_week5     2.253e-01  7.867e-02   2.864  0.00419 **
## duration         1.229e+00  2.319e-02  52.989  < 2e-16 ***
## campaign        -4.227e-02  1.407e-02  -3.004  0.00266 **
## previous         5.196e-02  6.790e-02   0.765  0.44409
## poutcome2        5.401e-01  1.143e-01   4.724 2.31e-06 ***
## poutcome3        1.846e+00  1.030e-01  17.918  < 2e-16 ***
## emp.var.rate    -1.720e+00  1.694e-01 -10.156  < 2e-16 ***
## cons.price.idx   1.245e+00  1.738e-01   7.161 8.03e-13 ***
## cons.conf.idx    1.027e-01  4.280e-02   2.399  0.01644 *
## euribor3m        5.056e-01  2.690e-01   1.880  0.06014 .
## nr.employed      3.791e-01  2.674e-01   1.418  0.15632
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 20299  on 28831  degrees of freedom
## Residual deviance: 11865  on 28785  degrees of freedom
## AIC: 11959
##
## Number of Fisher Scoring iterations: 10
```

```r
model1<-
glm(y~job+education+contact+month+day_of_week+duration+campaign+poutcome+emp.
var.rate+cons.price.idx+cons.conf.idx , family = binomial(link = 'logit') ,
data = train_set)
summary(model1) # aic = 11974
```

```
##
## Call:
## glm(formula = y ~ job + education + contact + month + day_of_week +
##     duration + campaign + poutcome + emp.var.rate + cons.price.idx +
##     cons.conf.idx, family = binomial(link = "logit"), data = train_set)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.9797  -0.2990  -0.1837  -0.1310   3.1406
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.27051    0.16456 -19.874  < 2e-16 ***
## job2           -0.27287    0.09269  -2.944 0.003239 **
## job3           -0.29353    0.15395  -1.907 0.056565 .
## job4            0.01177    0.17379   0.068 0.946005
## job5           -0.01168    0.10028  -0.117 0.907235
## job6            0.13733    0.10756   1.277 0.201665
## job7           -0.19361    0.14327  -1.351 0.176594
## job8           -0.15160    0.10318  -1.469 0.141779
## job9            0.09522    0.12427   0.766 0.443555
## job10          -0.01700    0.08466  -0.201 0.840878
```

```
## job11            0.11559    0.15042   0.768 0.442228
## education2       0.04904    0.14171   0.346 0.729281
## education3      -0.00833    0.11200  -0.074 0.940711
## education4       0.02710    0.10715   0.253 0.800340
## education5       1.12682    0.85397   1.320 0.187001
## education6       0.06173    0.11957   0.516 0.605683
## education7       0.22483    0.10377   2.167 0.030259 *
## contact2        -0.51334    0.08299  -6.186 6.17e-10 ***
## month2           0.68675    0.12328   5.571 2.54e-08 ***
## month3           0.38562    0.23443   1.645 0.099981 .
## month4           0.25204    0.11227   2.245 0.024778 *
## month5           0.01063    0.10745   0.099 0.921170
## month6           1.80673    0.13932  12.968  < 2e-16 ***
## month7          -0.53282    0.09051  -5.887 3.94e-09 ***
## month8          -0.24807    0.11530  -2.151 0.031441 *
## month9           0.25157    0.14418   1.745 0.081009 .
## month10          0.20485    0.15452   1.326 0.184930
## day_of_week2    -0.21880    0.08029  -2.725 0.006430 **
## day_of_week3     0.06326    0.07692   0.822 0.410841
## day_of_week4     0.07372    0.07919   0.931 0.351876
## day_of_week5     0.21933    0.07842   2.797 0.005159 **
## duration         1.22730    0.02319  52.935  < 2e-16 ***
## campaign        -0.04561    0.01413  -3.229 0.001243 **
## poutcome2        0.49834    0.07509   6.637 3.21e-11 ***
## poutcome3        1.84489    0.10194  18.097  < 2e-16 ***
## emp.var.rate    -0.99638    0.02885 -34.533  < 2e-16 ***
## cons.price.idx   0.75396    0.03936  19.157  < 2e-16 ***
## cons.conf.idx    0.10896    0.02827   3.854 0.000116 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 20299  on 28831  degrees of freedom
## Residual deviance: 11898  on 28794  degrees of freedom
## AIC: 11974
##
## Number of Fisher Scoring iterations: 6
```

2 Classification using Decision Tree

```
library(rpart)
classifier <- rpart(formula = y~., data = train_set)
```

3 Classification using SVM

```
library(e1071)
set.seed(123)
classifier1<-svm(formula = y~.,data = train_set, type = 'C-classification'
```

```
,kernel = 'radial')
classifier2<-svm(formula = y~.,data = train_set, type = 'C-classification'
,kernel = 'sigmoid')
classifier3<-svm(formula = y~.,data = train_set, type = 'C-classification'
,kernel = 'polynomial')
```

4.Classification using Naive Bayes classifier

```
set.seed(5465)
classifier4<-naiveBayes(y~.,data = train_set)
```

5.Classification using KNN

```
library(class)
y_pred7 <-knn(train = train_set,test = val_set,cl=train_set$y,k=300,prob =
TRUE)
```

6.Classification using Random Forest

```
set.seed(1234)
library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

classifier4 = randomForest(x=train_set[-20], y = train_set$y, ntree = 200)
```

Validation

1. Validating using logistic regression

```
# prediction
y_pred<-predict(model1 , newdata = val_set[-20])
y_pred = ifelse(y_pred>=0.5,1,0)
table(y_pred)

## y_pred
##     0     1
## 11745   611

write.csv(y_pred,file ="Final prediction.csv")
# confusion matrix
cm<-table(y_pred, val_set$y)
cm

##
## y_pred     0     1
```

```
##      0 10777   968
##      1   187   424

# accuracy
acc<- function(cm){
  tp<-cm[2,2]
  fp<-cm[2,1]
  acc<-tp/(tp+fp)
  acc
}
acc(cm) # accuracy = 69.39

## [1] 0.6939444
```

2.Validating using Decision Tree

```
# prediction
y_pred2<-predict(classifier , newdata = val_set[-20] , type = 'class')


# confusion matrix
cm2 = table(y_pred2 , val_set$ y)
cm2

##
## y_pred2     0     1
##       0 10495   631
##       1   469   761

# accuracy
acc(cm2)  # accuracy of DT is 61.86

## [1] 0.6186992
```

3.Validation using SVM

```
# prediction
y_pred3<-predict(classifier1, newdata = val_set[-20])
y_pred4<-predict(classifier2, newdata = val_set[-20])
y_pred5<-predict(classifier3, newdata = val_set[-20])

# confusion matrix
cm3<-table(y_pred3,val_set$y)
cm3

##
## y_pred3     0     1
##       0 10734   904
##       1   230   488
```

```
cm4<-table(y_pred4,val_set$y)
cm4

##
## y_pred4     0     1
##       0 10318   853
##       1   646   539

cm5<-table(y_pred5,val_set$y)
cm5

##
## y_pred5     0     1
##       0 10830  1023
##       1   134   369

# accuracy
acc(cm3) # accuracy is 67.96

## [1] 0.6796657

acc(cm4) # accuracy 45.48

## [1] 0.4548523

acc(cm5) # accuracy =73.35

## [1] 0.7335984
```

4.Validation using Naive Bayes classifier

```
# prediction
y_pred6<-predict(classifier4,newdata = val_set[-20])

#confusion matrix
cm6<-table(y_pred6,val_set$y)
cm6

##
## y_pred6     0     1
##       0 10565   679
##       1   399   713

# accuracy
acc(cm6) #accuracy = 64.11%

## [1] 0.6411871
```

5.Validation using KNN

```
#confusion matrix
cm7<-table(y_pred7,val_set$y)
cm7

##
## y_pred7     0     1
##       0 10956  1329
##       1     8    63

#accuracy
acc(cm7) #accuracy = 88.73%

## [1] 0.8873239
```

6.Validation using Random Forest

```
# prediction
y_pred8<-predict(classifier4,newdata = val_set[-20] )

# confusion matrix
cm8<- table(y_pred8,val_set$y)
cm8

##
## y_pred8     0     1
##       0 10566   674
##       1   398   718

# accuracy
acc(cm8)  # accuracy = 64.33%

## [1] 0.6433692
```

Testing

Reading the Test Data

```
test_set<-read.csv('D:/IMARTICUS/final project/bank-additional.csv',na.string
= c(""," ","NA"))
```

Imputaion

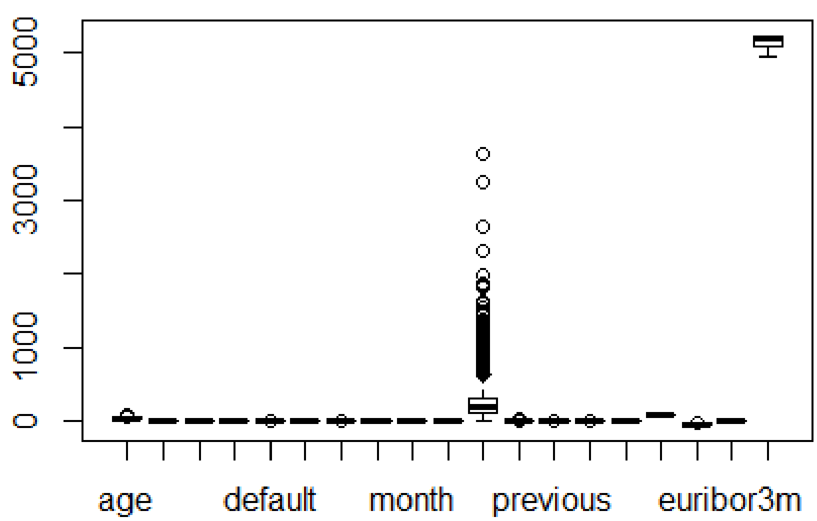
```
#checking the NA's in the test_set
colSums(is.na(test_set))

##            age            job        marital      education        default
##              0              0              0              0              0
```

```
##        housing           loan        contact          month    day_of_week
##              0              0              0              0              0
##       duration       campaign       previous       poutcome   emp.var.rate
##              0              0              0              0              0
## cons.price.idx  cons.conf.idx      euribor3m    nr.employed
##              0              0              0              0
```

```r
#checking the dimensions of test_set
dim(test_set)
```

```
## [1] 3090   19
```

```r
# checking for the outliers
boxplot(test_set)
```




```r
# scaling the dataset
test_set$duration<-scale(test_set$duration)
test_set$age<-scale(test_set$age)
test_set$cons.price.idx<-scale(test_set$cons.price.idx)
test_set$cons.conf.idx<-scale(test_set$cons.conf.idx)
test_set$euribor3m<-scale(test_set$euribor3m)
test_set$nr.employed<-scale(test_set$nr.employed)

# converting data into levels

levels(test_set$job)<-1:11
levels(test_set$marital)<-1:3
```

```r
levels(test_set$education)<-1:7
levels(test_set$default)<-1:2
levels(test_set$housing)<-1:2
levels(test_set$loan)<-1:2
levels(test_set$contact)<-1:2
levels(test_set$month)<-1:10
levels(test_set$day_of_week)<-1:5
levels(test_set$poutcome)<-1:3
```

Prediction on test data

After Model building and validation I found that **KNN** is the best machine learning approach for this dataset.

Prediction using KNN

```r
result <-knn(train = train_set[-20],test = test_set,cl=train_set$y,k=300,prob
= TRUE)
test_set<-data.frame(test_set,result)
summary(test_set)
```

```
##        age.V1              job        marital    education default   housing
##  Min.   :-1.883753   1       :854   1: 348   1: 243    1:3089   1:1402
##  1st Qu.:-0.803627   10      :573   2:1791   2: 150    2:   1   2:1688
##  Median :-0.214468   2       :554   3: 951   3: 407
##  Mean   : 0.000000   8       :276            4: 728
##  3rd Qu.: 0.669271   5       :265            5:   1
##  Max.   : 4.793387   7       :126            6: 454
##                      (Other):442             7:1107
##  loan     contact      month      day_of_week     duration.V1
##  1:2583   1:2108   7       :981   1:580        Min.   :-0.982137
##  2: 507   2: 982   4       :514   2:642        1st Qu.:-0.588138
##                    2       :495   3:630        Median :-0.296427
##                    8       :387   4:613        Mean   : 0.000000
##                    5       :365   5:625        3rd Qu.: 0.211225
##                    1       :169                Max.   :12.819186
##                    (Other):179
##     campaign         previous       poutcome  emp.var.rate
##  Min.   : 1.000   Min.   :0.0000   1: 360   Min.   :-3.4000
##  1st Qu.: 1.000   1st Qu.:0.0000   2:2602   1st Qu.:-1.8000
##  Median : 2.000   Median :0.0000   3: 128   Median : 1.1000
##  Mean   : 2.509   Mean   :0.2081            Mean   :-0.0468
##  3rd Qu.: 3.000   3rd Qu.:0.0000            3rd Qu.: 1.4000
##  Max.   :35.000   Max.   :6.0000            Max.   : 1.4000
##
##   cons.price.idx.V1     cons.conf.idx.V1       euribor3m.V1
##  Min.   :-2.2721540   Min.   :-2.1450550   Min.   :-1.6082939
##  1st Qu.:-0.7803963   1st Qu.:-0.4385156   1st Qu.:-1.2252465
##  Median :-0.1505810   Median :-0.2489002   Median : 0.7764304
```

```
##   Mean   : 0.0000000   Mean   : 0.0000000   Mean   : 0.0000000
##   3rd Qu.: 0.7881681   3rd Qu.: 0.8887928   3rd Qu.: 0.8357519
##   Max.   : 2.1075373   Max.   : 2.8902895   Max.   : 0.8832091
##
##     nr.employed.V1    result
##   Min.   :-2.5828077   0:3074
##   1st Qu.:-0.8111527   1:   16
##   Median : 0.3904348
##   Mean   : 0.0000000
##   3rd Qu.: 0.8755152
##   Max.   : 0.8755152
##
```

```r
summary(result)
```

```
##    0    1
## 3074   16
```

```r
Write Test Result CSV in R


write.csv(test_set, file = "Final_Test_Result.csv")
```

Questions:

1. Which machine learning approach is appropriate to find the solution for the above mentioned problem?

   We found out that knn is the best machine learning approach for this dataset.

2. Predict the term deposit subscription for the Bank additional dataset and conclude if the telemarketing campaign was a success or not.

   After we run summary on the result variable we came to know that there are only 15 people out of 3090 people who secured the term deposit .Thus we can conclude that the telemarketing campaign was an utter failure.

3. What is are the key differentiators between the ones who have subscribed (Yes) and who did not (No).

   As we have used knn machine learning algorith we actually don’t know which are the significant variables effecting the prediction process.