Building a robust Geodemographic Segmentation Model

Applying Logistic Regression and step by step building a model

Grouping of customers by similarities of their behavior and using prior knowledge to predict any future trends and basically predict future behavior. Here, we did Churn Modeling to understand when your customers are going to leave and who's more likely to leave, who's less likely to leave in a bank scenario.

```
#Geodemographic segmentation model

#Complete data set
library(readxl)
Churn_Modelling <- read_excel("C:/Users/bvkka/Desktop/Udemy/Data Science/Churn-Modelling.xlsx")
View(Churn_Modelling)




# creating dummy variables
install.packages("plyr")
library(plyr)
Churn_Modelling$Geography <- revalue(Churn_Modelling$Geography,c("France"=0))
Churn_Modelling$Geography <- revalue(Churn_Modelling$Geography,c("Spain"=1))
Churn_Modelling$Geography <- revalue(Churn_Modelling$Geography,c("Germany"=2))


Churn_Modelling$Gender <- revalue(Churn_Modelling$Gender,c("Female"=0))
Churn_Modelling$Gender <- revalue(Churn_Modelling$Gender,c("Male"=1))

Female<-as.numeric(Churn_Modelling$Gender==0)
Spain<-as.numeric(Churn_Modelling$Geography==1)
Germany<-as.numeric(Churn_Modelling$Geography==2)


#character to numeric
Churn_Modelling$Gender<-as.numeric(as.character(Churn_Modelling$Gender))
Churn_Modelling$Geography<-as.numeric(as.character(Churn_Modelling$Geography))
```

```r
wealthAccumulation<-(Churn_Modelling$Balance)/(Churn_Modelling$Age)
Age1<-Churn_Modelling$Age
Balance1<-Churn_Modelling$Balance
Balance2<-log10(Balance1+1)
wealthAccumulationlog<-log10(Balance1/Age1+1)

#logestic regression
#creating a model
model1<-glm(formula = Exited ~ CreditScore+wealthAccumulationlog+Age+NumOfProducts+IsActiveMember+Female+Germany,
binomial(link="logit"),data =Churn_Modelling)
summary(model1)
#library(dplyr)
prob_predict1=predict(model1,type = 'response')

summary(prob_predict1)


y_pred=ifelse(prob_predict1>0.5,1,0) #vector of predictions
y_pred
#Making the confusion Matrix
cm=table(y_pred,Churn_Modelling$Exited)
cm
TN<-cm[1] #7681
FN<-cm[2] #282
FP<-cm[3] #1605
TP<-cm[4] #432

ActualYes<-FN+TP
ActualNo<-TN+FP
PredictedYes<-FP+TP
PredictedNo<-TN+FN
#accuracy (TP+TN)/(TP+TN+FN+FP)
accuracy1=mean(y_pred==Churn_Modelling$Exited)
accuracy1
#misclassification rate: (FP+FN)/(TP+TN+FN+FP)
MR<-(FP+FN)/(TN+FN+FP+TP)
```

```
#TPR : When it's actually yes, how often does it predict yes?
TPR<-TP/(ActualYes)
#FPR : when it's actually no, how often does it predict yes?
FPR<-FP/(ActualNo)
#precision : when it predicts yes, how often is it correct? TP/TP+FP
precision1=precision<-diag(cm)/colSums(cm)
precision1
#recall :
recall1=recall<-diag(cm)/rowSums(cm)
recall1
#ROC Curve
library(pROC)
myROC<-roc(response=Churn_Modelling$Exited,predictor = prob_predict1,positve='prob_predict1')
plot(myROC)
pred1<-prediction(prob_predict1,Churn_Modelling$Exited)
roc.perf=performance(pred1,measure = "tpr",x.measure = "fpr")
ggplot(mode)


#auc 0.7669
auc(roc(Churn_Modelling$Exited,prob_predict1))


#####now we add a new test data and see how classifier predicts#####

#test data set
##merged test data with train data except the lasr column, model should predict that

library(readxl)
Churn_Modelling_testt <- read_excel("C:/Users/bvkka/Desktop/Udemy/Data Science/Churn-Modelling-testt.xlsx")
View(Churn_Modelling_testt)


Churn_Modelling_testt$Gender <- revalue(Churn_Modelling_testt$Gender,c("Female"=0))
Churn_Modelling_testt$Gender <- revalue(Churn_Modelling_testt$Gender,c("Male"=1))
Churn_Modelling_testt$Geography <- revalue(Churn_Modelling_testt$Geography,c("France"=0))
Churn_Modelling_testt$Geography <- revalue(Churn_Modelling_testt$Geography,c("Spain"=1))
Churn_Modelling_testt$Geography <- revalue(Churn_Modelling_testt$Geography,c("Germany"=2))
```

```
Churn_Modelling_testt$Gender<-as.numeric(as.character(Churn_Modelling_testt$Gender))
Churn_Modelling_testt$Geography<-as.numeric(as.character(Churn_Modelling_testt$Geography))

model2<-glm(formula = Exited ~ CreditScore+wealthAccumulationlog+Age+NumOfProducts+IsActiveMember+Female+Germany,
binomial(link="logit"),data =Churn_Modelling_testt)
summary(model2)
#predicting the test set results
prob_pred2=predict(model1,type='response',newdata = Churn_Modelling_testt)   #for predicitng we only need
predictors , but not response
summary(prob_pred2)
y_pred2=ifelse(prob_pred2>0.5,1,0) #vector of predictions
y_pred2

#Making the confusion Matrix
cm=table(y_pred2,Churn_Modelling_testt$Exited)
cm
```

1. <u>Building the model – First iteration</u>

```
> summary(model1)

Call:
glm(formula = Exited ~ CreditScore + Age + Tenure + Balance +
    NumOfProducts + HasCrCard + IsActiveMember + EstimatedSalary +
    Female + Spain + Germany, family = binomial(link = "logit"),
    data = Churn_Modelling)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3097  -0.6589  -0.4560  -0.2697   2.9940

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -3.921e+00  2.454e-01 -15.980  < 2e-16 ***
CreditScore     -6.683e-04  2.803e-04  -2.384   0.0171 *
Age              7.271e-02  2.576e-03  28.230  < 2e-16 ***
Tenure          -1.595e-02  9.355e-03  -1.705   0.0882 .
Balance          2.637e-06  5.142e-07   5.128 2.92e-07 ***
NumOfProducts   -1.015e-01  4.713e-02  -2.154   0.0312 *
HasCrCard       -4.468e-02  5.934e-02  -0.753   0.4515
IsActiveMember  -1.075e+00  5.769e-02 -18.643  < 2e-16 ***
EstimatedSalary  4.807e-07  4.737e-07   1.015   0.3102
Female           5.285e-01  5.449e-02   9.699  < 2e-16 ***
Spain            3.522e-02  7.064e-02   0.499   0.6181
Germany          7.747e-01  6.767e-02  11.448  < 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: 10109.8  on 9999  degrees of freedom
Residual deviance:  8561.4  on 9988  degrees of freedom
AIC: 8585.4

Number of Fisher Scoring iterations: 5
```

2. <u>Applying backward elimination: step by step</u>
   Now checking the p values, we see that p value of Country: Spain is very high (0.6181)
   Greater than threshold(p=0.05), so we are excluding Spain.
   2nd Iteration:

```
Call:
glm(formula = Exited ~ CreditScore + Age + Tenure + Balance + 
    NumOfProducts + HasCrCard + IsActiveMember + EstimatedSalary + 
    Female + Germany, family = binomial(link = "logit"), data = Churn_Modelling)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.3099  -0.6584  -0.4559  -0.2691   2.9901  

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)    
(Intercept)     -3.911e+00  2.445e-01 -15.994  < 2e-16 ***
CreditScore     -6.666e-04  2.803e-04  -2.378   0.0174 *  
Age              7.272e-02  2.575e-03  28.238  < 2e-16 ***
Tenure          -1.598e-02  9.354e-03  -1.708   0.0876 .  
Balance          2.637e-06  5.142e-07   5.129 2.91e-07 ***
NumOfProducts   -1.013e-01  4.713e-02  -2.149   0.0316 *  
HasCrCard       -4.493e-02  5.934e-02  -0.757   0.4489    
IsActiveMember  -1.075e+00  5.768e-02 -18.640  < 2e-16 ***
EstimatedSalary  4.813e-07  4.736e-07   1.016   0.3095    
Female           5.283e-01  5.449e-02   9.697  < 2e-16 ***
Germany          7.629e-01  6.336e-02  12.041  < 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: 10109.8  on 9999  degrees of freedom
Residual deviance:  8561.6  on 9989  degrees of freedom
AIC: 8583.6

Number of Fisher Scoring iterations: 5
```

Now checking the p values, we see that p value of HasCrCard is very high (0.4489)
Greater than threshold(p=0.05), so we are excluding that as well.
3rd Iteration:

```
Call:
glm(formula = Exited ~ CreditScore + Age + Tenure + Balance +
    NumOfProducts + IsActiveMember + EstimatedSalary + Female +
    Germany, family = binomial(link = "logit"), data = Churn_Modelling)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3152  -0.6585  -0.4565  -0.2699   2.9859

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)     -3.944e+00  2.406e-01 -16.395  < 2e-16 ***
CreditScore     -6.640e-04  2.803e-04  -2.369   0.0178 *
Age              7.273e-02  2.575e-03  28.243  < 2e-16 ***
Tenure          -1.615e-02  9.351e-03  -1.727   0.0842 .
Balance          2.645e-06  5.141e-07   5.146 2.66e-07 ***
NumOfProducts   -1.013e-01  4.712e-02  -2.150   0.0315 *
IsActiveMember  -1.074e+00  5.767e-02 -18.631  < 2e-16 ***
EstimatedSalary  4.818e-07  4.737e-07   1.017   0.3091
Female           5.285e-01  5.449e-02   9.700  < 2e-16 ***
Germany          7.619e-01  6.334e-02  12.028  < 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: 10109.8  on 9999  degrees of freedom
Residual deviance:  8562.2  on 9990  degrees of freedom
AIC: 8582.2

Number of Fisher Scoring iterations: 5
```

Now checking the p values, we see that p value of EstimatedSalary is high (0.3091) Greater than threshold(p=0.05), so we are excluding that as well.

4^{th} Iteration:

```
Call:
glm(formula = Exited ~ CreditScore + Age + Tenure + Balance + 
    NumOfProducts + IsActiveMember + Female + Germany, family = binomial(link = "log
it"), 
    data = Churn_Modelling)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.3272  -0.6592  -0.4557  -0.2688   2.9787  

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)    
(Intercept)    -3.896e+00  2.357e-01 -16.528  < 2e-16 ***
CreditScore    -6.664e-04  2.803e-04  -2.378   0.0174 *  
Age             7.270e-02  2.575e-03  28.238  < 2e-16 ***
Tenure         -1.598e-02  9.349e-03  -1.710   0.0873 .  
Balance         2.653e-06  5.140e-07   5.162 2.44e-07 ***
NumOfProducts  -1.005e-01  4.712e-02  -2.132   0.0330 *  
IsActiveMember -1.075e+00  5.766e-02 -18.644  < 2e-16 ***
Female          5.290e-01  5.448e-02   9.710  < 2e-16 ***
Germany         7.621e-01  6.334e-02  12.031  < 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: 10109.8  on 9999  degrees of freedom
Residual deviance:  8563.2  on 9991  degrees of freedom
AIC: 8581.2

Number of Fisher Scoring iterations: 5
```

Now checking the p values, we see that p value of Tenure is high (0.0873)
Greater than threshold(p=0.05), so we are excluding that as well.
5th Iteration:

```
Call:
glm(formula = Exited ~ CreditScore + Age + Balance + NumOfProducts +
    IsActiveMember + Female + Germany, family = binomial(link = "logit"),
    data = Churn_Modelling)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3330  -0.6574  -0.4560  -0.2697   2.9674

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)    -3.976e+00  2.312e-01 -17.200  < 2e-16 ***
CreditScore    -6.660e-04  2.802e-04  -2.377   0.0175 *
Age             7.269e-02  2.574e-03  28.237  < 2e-16 ***
Balance         2.652e-06  5.139e-07   5.160 2.46e-07 ***
NumOfProducts  -1.010e-01  4.709e-02  -2.144   0.0320 *
IsActiveMember -1.072e+00  5.761e-02 -18.603  < 2e-16 ***
Female          5.306e-01  5.447e-02   9.741  < 2e-16 ***
Germany         7.608e-01  6.333e-02  12.014  < 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: 10109.8  on 9999  degrees of freedom
Residual deviance:  8566.1  on 9992  degrees of freedom
AIC: 8582.1

Number of Fisher Scoring iterations: 5
```

For R squared – use library rcompanion and also finding ANOVA for the model

```
> library(rcompanion)
Warning message:
package 'rcompanion' was built under R version 3.3.3
> nagelkerke((model1.final))
Error in nagelkerke((model1.final)) : object 'model1.final' not found
> nagelkerke(model1)
$Models


Model: "glm, Exited ~ CreditScore + Age + Balance + NumOfProducts + IsActiveMember + Female + G
ermany, binomial(link = \"logit\"), Churn_Modelling"
Null:  "glm, Exited ~ 1, binomial(link = \"logit\"), Churn_Modelling"


$Pseudo.R.squared.for.model.vs.null
                             Pseudo.R.squared
McFadden                             0.152689
Cox and Snell (ML)                   0.143041
Nagelkerke (Cragg and Uhler)         0.224858

$Likelihood.ratio.test
 Df.diff LogLik.diff  Chisq p.value
      -7     -771.82 1543.6       0

$Number.of.observations

Model: 10000
Null:  10000

$Messages
[1] "Note: For models fit with REML, these statistics are based on refitting with ML"

$Warnings
[1] "None"

> anova(model1)
Analysis of Deviance Table

Model: binomial, link: logit

Response: Exited

Terms added sequentially (first to last)


               Df Deviance Resid. Df Resid. Dev
NULL                            9999    10109.8
CreditScore     1     7.34      9998    10102.4
Age             1   759.17      9997     9343.3
Balance         1   139.41      9996     9203.9
NumOfProducts   1     0.58      9995     9203.3
IsActiveMember  1   390.04      9994     8813.2
Female          1   102.93      9993     8710.3
Germany         1   144.19      9992     8566.1
```

Coefficients and exponential coefficients

At 95% CI

```
> confint(model1)
Waiting for profiling to be done...
                       2.5 %        97.5 %
(Intercept)    -4.431022e+00 -3.524759e+00
CreditScore    -1.215501e-03 -1.169776e-04
Age             6.766748e-02  7.775953e-02
Balance         1.644340e-06  3.658953e-06
NumOfProducts  -1.936097e-01 -8.979007e-03
IsActiveMember -1.185278e+00 -9.594020e-01
Female          4.239580e-01  6.374826e-01
Germany         6.367768e-01  8.850554e-01
> exp(confint(model1))
Waiting for profiling to be done...
                    2.5 %     97.5 %
(Intercept)    0.01190233 0.02945891
CreditScore    0.99878524 0.99988303
Age            1.07000945 1.08086271
Balance        1.00000164 1.00000366
NumOfProducts  0.82397941 0.99106118
IsActiveMember 0.30566111 0.38312194
Female         1.52799741 1.89171274
Germany        1.89037789 2.42311858
> 
```

3. <u>Step 3: Applying significant transformations to the independent variables:</u> to get better results or make the model more robust: We are going to change variable ‘balance’ to ‘log10(balance+1)’ i.e applying logarithmic transformation to the varaiable and then run the model again

```
> summary(model1)

Call:
glm(formula = Exited ~ CreditScore + Age + log10(Balance + 1) + 
    NumOfProducts + IsActiveMember + Female + Germany, family = binomial(link = "logit"), 
    data = Churn_Modelling)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.3162  -0.6582  -0.4574  -0.2694   2.9716  

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)    
(Intercept)        -3.9923842  0.2326151 -17.163  < 2e-16 ***
CreditScore        -0.0006744  0.0002802  -2.407   0.0161 *  
Age                 0.0726405  0.0025741  28.220  < 2e-16 ***
log10(Balance + 1)  0.0690313  0.0139553   4.947 7.55e-07 ***
NumOfProducts      -0.0954940  0.0475088  -2.010   0.0444 *  
IsActiveMember     -1.0725273  0.0575976 -18.621  < 2e-16 ***
Female              0.5283013  0.0544440   9.704  < 2e-16 ***
Germany             0.7463025  0.0650378  11.475  < 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: 10110  on 9999  degrees of freedom
Residual deviance:  8568  on 9992  degrees of freedom
AIC: 8584

Number of Fisher Scoring iterations: 5
```

We notice that the p value of balance (After log transformation) has become more significant.

```
> nagelkerke(model1)
$Models

Model: "glm, Exited ~ CreditScore + Age + log10(Balance + 1) + NumOfProducts + IsActiveMember +
 Female + Germany, binomial(link = \"logit\"), Churn_Modelling"
Null:  "glm, Exited ~ 1, binomial(link = \"logit\"), Churn_Modelling"

$Pseudo.R.squared.for.model.vs.null
                             Pseudo.R.squared
McFadden                             0.152501
Cox and Snell (ML)                   0.142878
Nagelkerke (Cragg and Uhler)         0.224603

$Likelihood.ratio.test
 Df.diff LogLik.diff  Chisq p.value
      -7     -770.88 1541.8       0

$Number.of.observations

Model: 10000
Null:  10000

$Messages
[1] "Note: For models fit with REML, these statistics are based on refitting with ML"

$Warnings
[1] "None"

>
```

But now R squared dropped a bit. Nevertheless, I prefer to keep the balance as log balance, we don't always get such results. Sometimes it happens, good thing is accuracy increased.

4. <u>Creating a derived variable:</u>

Wealth Accumulation: Balance/Age

```
Call:
glm(formula = Exited ~ CreditScore + wealthAccumulation + Age +
    log10(Balance + 1) + NumOfProducts + IsActiveMember + Female +
    Germany, family = binomial(link = "logit"), data = Churn_Modelling)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3107  -0.6584  -0.4555  -0.2694   2.9550

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)        -3.908e+00  2.438e-01 -16.031  < 2e-16 ***
CreditScore        -6.751e-04  2.803e-04  -2.409 0.016014 *
wealthAccumulation -4.308e-05  3.778e-05  -1.140 0.254187
Age                 7.067e-02  3.094e-03  22.839  < 2e-16 ***
log10(Balance + 1)  9.494e-02  2.662e-02   3.567 0.000361 ***
NumOfProducts      -9.600e-02  4.753e-02  -2.020 0.043397 *
IsActiveMember     -1.070e+00  5.762e-02 -18.571  < 2e-16 ***
Female              5.273e-01  5.446e-02   9.683  < 2e-16 ***
Germany             7.450e-01  6.512e-02  11.441  < 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: 10109.8  on 9999  degrees of freedom
Residual deviance:  8566.7  on 9991  degrees of freedom
AIC: 8584.7

Number of Fisher Scoring iterations: 5
```

Hmmm now this new variable is not significant at all, even r square decreased. But, we are going to remove other variables to make this significant. We will see that in multicollinearity using VIF.

5. <u>Checking for multicollinearity using VIF:</u>

Sometimes independent variables are corelated and can damage the model. So, here wealth accumulation, age and balance are all linked. Do VIF and remove the variable with VIF > 5.

```
> library(car)
Warning message:
package 'car' was built under R version 3.3.3
> vif(model1)
      CreditScore wealthAccumulation                Age log10(Balance + 1)
         1.001029           5.186575           1.555937           5.017300
    NumOfProducts     IsActiveMember             Female            Germany
         1.099532           1.077147           1.003326           1.284374
```

Here, you can see wealth acc and balance having vif>5. Lets go ahead and remove Balance variable and then run the model.

```
Call:
glm(formula = Exited ~ CreditScore + wealthAccumulation + Age +
    NumOfProducts + IsActiveMember + Female + Germany, family = binomial(link = "logit"),
    data = Churn_Modelling)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3283  -0.6582  -0.4571  -0.2719   2.9773

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)        -4.014e+00  2.419e-01 -16.591  < 2e-16 ***
CreditScore        -6.714e-04  2.800e-04  -2.398 0.016493 *
wealthAccumulation  7.084e-05  1.945e-05   3.641 0.000271 ***
Age                 7.582e-02  2.749e-03  27.584  < 2e-16 ***
NumOfProducts      -1.214e-01  4.708e-02  -2.579 0.009905 **
IsActiveMember     -1.076e+00  5.762e-02 -18.668  < 2e-16 ***
Female              5.279e-01  5.441e-02   9.701  < 2e-16 ***
Germany             8.068e-01  6.291e-02  12.824  < 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: 10109.8  on 9999  degrees of freedom
Residual deviance:  8579.5  on 9992  degrees of freedom
AIC: 8595.5

Number of Fisher Scoring iterations: 5
```

We can notice the wealth accumulation variable is now very significant.

Now let us try making another transformation. Lets do log of both balance and Wealth Accumulation and include them in the model and run again.

```
Churn_Modelling$Gender<-as.numeric(as.character(Churn_Modelling$Gender))
Churn_Modelling$Geography<-as.numeric(as.character(Churn_Modelling$Geography))

Churn_Modelling_Test_Data$Gender<-as.numeric(as.character(Churn_Modelling_Test_Data$Gende
Churn_Modelling_Test_Data$Geography<-as.numeric(as.character(Churn_Modelling_Test_Data$Ge

wealthAccumulation<-(Churn_Modelling$Balance)/(Churn_Modelling$Age)
Age1<-Churn_Modelling$Age
Balance1<-Churn_Modelling$Balance
wealthAccumulationlog<-log10(Balance1/Age1+1)

#logestic regression
#creating a model
model1<-glm(formula = Exited ~ CreditScore+wealthAccumulationlog+log10(Balance + 1)+Age+N
summary(model1)
```

We notice that both Wealth accumulation log and Balance log variables are significant in this model now.

```
Call:
glm(formula = Exited ~ CreditScore + wealthAccumulationlog + 
    log10(Balance + 1) + Age + NumOfProducts + IsActiveMember + 
    Female + Germany, family = binomial(link = "logit"), data = Churn_Modelling)

Deviance Residuals: 
    Min       1Q   Median       3Q      Max  
-2.3304  -0.6560  -0.4551  -0.2731   2.9053  

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)    
(Intercept)           -3.6572945  0.2631004 -13.901  < 2e-16 ***
CreditScore           -0.0006689  0.0002805  -2.385  0.01709 *  
wealthAccumulationlog -1.1675855  0.4351260  -2.683  0.00729 ** 
log10(Balance + 1)     0.8640397  0.2966410   2.913  0.00358 ** 
Age                    0.0647517  0.0038898  16.647  < 2e-16 ***
NumOfProducts         -0.0987752  0.0476219  -2.074  0.03807 *  
IsActiveMember        -1.0655556  0.0576719 -18.476  < 2e-16 ***
Female                 0.5279634  0.0544969   9.688  < 2e-16 ***
Germany                0.7502325  0.0653833  11.474  < 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: 10109.8  on 9999  degrees of freedom
Residual deviance:  8560.8  on 9991  degrees of freedom
AIC: 8578.8

Number of Fisher Scoring iterations: 5
```

We also note that R squared value is increased and better than previous ones

```
> nagelkerke(model1)
$Models

Model: "glm, Exited ~ CreditScore + wealthAccumulationlog + log10(Balance + 1) + Age + NumOfPro
ducts + IsActiveMember + Female + Germany, binomial(link = \"logit\"), Churn_Modelling"
Null:  "glm, Exited ~ 1, binomial(link = \"logit\"), Churn_Modelling"

$Pseudo.R.squared.for.model.vs.null
                             Pseudo.R.squared
McFadden                             0.153216
Cox and Snell (ML)                   0.143497
Nagelkerke (Cragg and Uhler)         0.225576

$Likelihood.ratio.test
 Df.diff LogLik.diff Chisq p.value
      -8     -774.49  1549       0

$Number.of.observations

Model: 10000
Null:  10000

$Messages
[1] "Note: For models fit with REML, these statistics are based on refitting with ML"

$Warnings
[1] "None"
```

But, we cannot conclude that the model is robust now. Check VIF now

```
> vif(model1)
        CreditScore wealthAccumulationlog    log10(Balance + 1)                   Age
           1.001055            630.623050            627.435235              2.438170
      NumOfProducts        IsActiveMember                Female               Germany
           1.101203              1.077199              1.003157              1.288809
>
```

The values of WA and balance are through the roof, so it means these both are the same, extra ordinary collinearity. So, we have to omit one which makes the model robust, p values significant and R squared values high, So by log balance out and check

```
Call:
glm(formula = Exited ~ CreditScore + wealthAccumulationlog +
    Age + NumOfProducts + IsActiveMember + Female + Germany,
    family = binomial(link = "logit"), data = Churn_Modelling)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3157  -0.6581  -0.4567  -0.2696   2.9756

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)           -4.0122516  0.2342851 -17.126  < 2e-16 ***
CreditScore           -0.0006747  0.0002802  -2.408   0.0160 *
wealthAccumulationlog  0.0985733  0.0204716   4.815 1.47e-06 ***
Age                    0.0733040  0.0025813  28.398  < 2e-16 ***
NumOfProducts         -0.0971785  0.0475229  -2.045   0.0409 *
IsActiveMember        -1.0730589  0.0575960 -18.631  < 2e-16 ***
Female                 0.5281740  0.0544377   9.702  < 2e-16 ***
Germany                0.7502080  0.0650527  11.532  < 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: 10109.8  on 9999  degrees of freedom
Residual deviance:  8569.3  on 9992  degrees of freedom
AIC: 8585.3

Number of Fisher Scoring iterations: 5
```

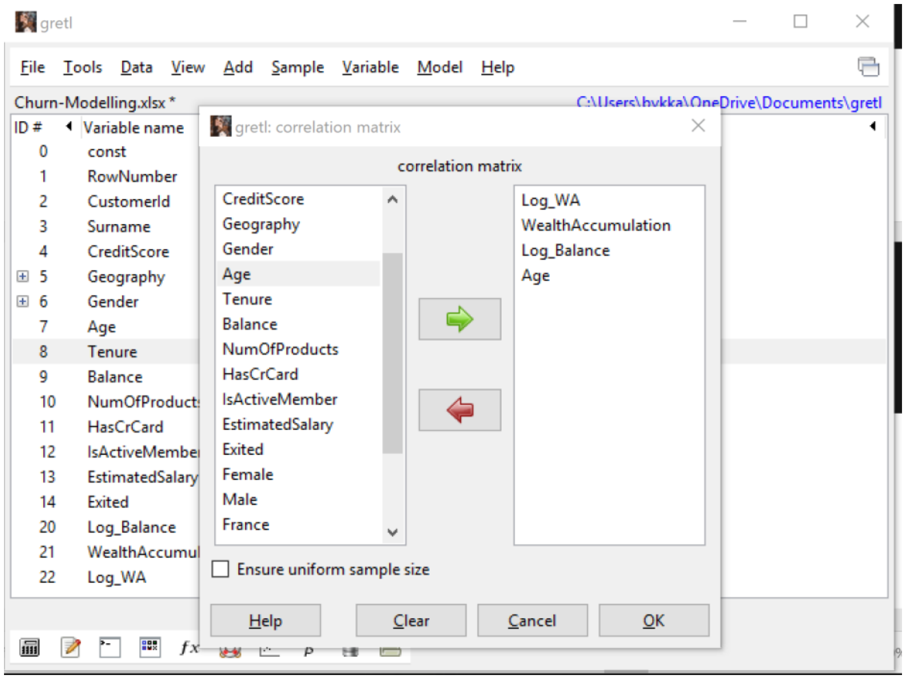
```
> nagelkerke(model1)
$Models

Model: "glm, Exited ~ CreditScore + wealthAccumulationlog + Age + NumOfProducts + IsActiveMemb
r + Female + Germany, binomial(link = \"logit\"), Churn_Modelling"
Null:  "glm, Exited ~ 1, binomial(link = \"logit\"), Churn_Modelling"


$Pseudo.R.squared.for.model.vs.null
                             Pseudo.R.squared
McFadden                             0.152373
Cox and Snell (ML)                   0.142768
Nagelkerke (Cragg and Uhler)         0.224429

$Likelihood.ratio.test
 Df.diff LogLik.diff  Chisq p.value
      -7     -770.23 1540.5       0

$Number.of.observations

Model: 10000
Null:  10000

$Messages
[1] "Note: For models fit with REML, these statistics are based on refitting with ML"

$Warnings
[1] "None"

>
```

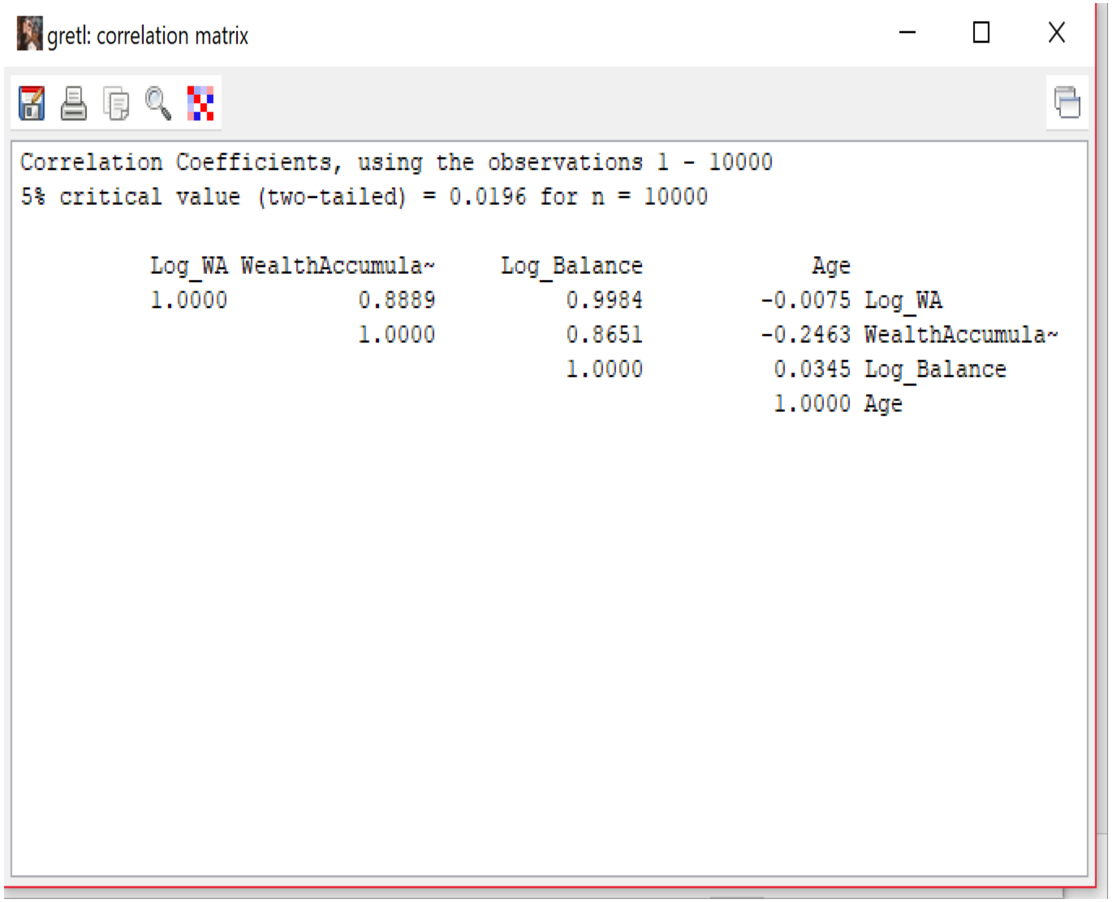
```
vif(model1)
          CreditScore wealthAccumulationlog                   Age         NumOfProducts
             1.001031              1.388246              1.087915              1.099674
       IsActiveMember                Female               Germany
             1.076231              1.003092              1.284242
```

Everything satisfied now ☺

6. Correction Matrix: a **correlation matrix**, which is used to investigate the dependence between multiple variables at the same time. The result is a table containing the **correlation coefficients**between each variable and the others.

I'm showing using gretl






```
Correlation Coefficients, using the observations 1 - 10000
5% critical value (two-tailed) = 0.0196 for n = 10000

         Log_WA WealthAccumula~      Log_Balance            Age
         1.0000          0.8889           0.9984        -0.0075 Log_WA
                         1.0000           0.8651        -0.2463 WealthAccumula~
                                          1.0000         0.0345 Log_Balance
                                                         1.0000 Age
```

You can see log balance and log wa are almost same, therefore bad for our model and we had to remove one.

7. <u>Final trained model:</u>

gretl: model 2

File Edit Tests Save Graphs Analysis LaTeX

```
Model 2: Logit, using observations 1-10000
Dependent variable: Exited
Standard errors based on Hessian

                   coefficient    std. error       z       p-value
  -----------------------------------------------------------------
  const            -3.91258       0.237164      -16.50    3.84e-061 ***
  CreditScore      -0.000674866   0.000280272    -2.408   0.0160    **
  Germany           0.747595      0.0650515      11.49    1.44e-030 ***
  Female            0.526721      0.0544591       9.672   3.97e-022 ***
  Age               0.0726550     0.00257451     28.22    3.24e-175 ***
  NumOfProducts    -0.0950198     0.0475374      -1.999   0.0456    **
  IsActiveMember   -1.07578       0.0576458     -18.66    1.01e-077 ***
  Log_Balance       0.0690263     0.0139592       4.945   7.62e-07  ***
  Tenure           -0.0158791     0.00934627     -1.699   0.0893    *

Mean dependent var   0.203700   S.D. dependent var   0.402769
McFadden R-squared   0.152787   Adjusted R-squared   0.151006
Log-likelihood      -4282.570   Akaike criterion     8583.141
Schwarz criterion    8648.034   Hannan-Quinn         8605.107

Number of cases 'correctly predicted' = 8127 (81.3%)
f(beta'x) at mean of independent vars = 0.135
Likelihood ratio test: Chi-square(8) = 1544.64 [0.0000]

           Predicted
               0      1
  Actual 0  7687    276
         1  1597    440
```

By looking at thecoefficeints,

Germany people are more likely to leave the bank.

Odds ratio we have to calculate

Female customers are more likely to leave

```
> #library(dplyr)
> prob_predict1=predict(model1,type = 'response')
> prob_predict1=as.numeric(predict1)
> summary(prob_predict1)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.01094 0.08142 0.15350 0.20370 0.27550 0.93150
>
> y_pred=ifelse(prob_predict1>0.5,1,0) #vector of predictions
> y_pred
   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0
  [93] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0
 [185] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [277] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
 [369] 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [461] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 [553] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
 [645] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0
 [737] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0
 [829] 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [921] 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [ reached getOption("max.print") -- omitted 9000 entries ]
> #Making the confusion Matrix
> cm=table(y_pred,Churn_Modelling$Exited)
> cm

y_pred    0    1
     0 7681 1605
     1  282  432
```

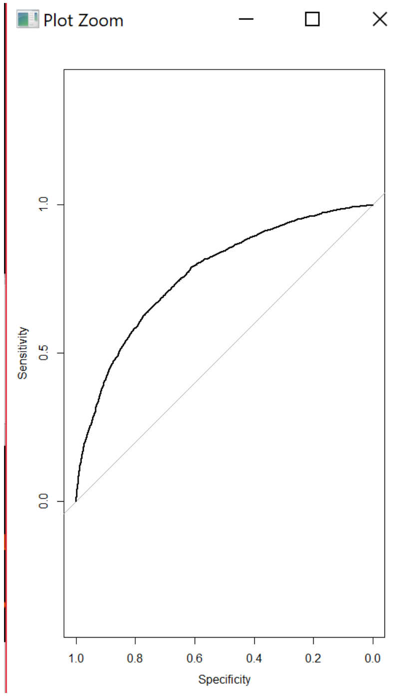
```
> accuracy1=mean(y_pred==Churn_Modelling$Exited)
> accuracy1
[1] 0.8113
>
```

1605+432 = 2037 = Total Exited

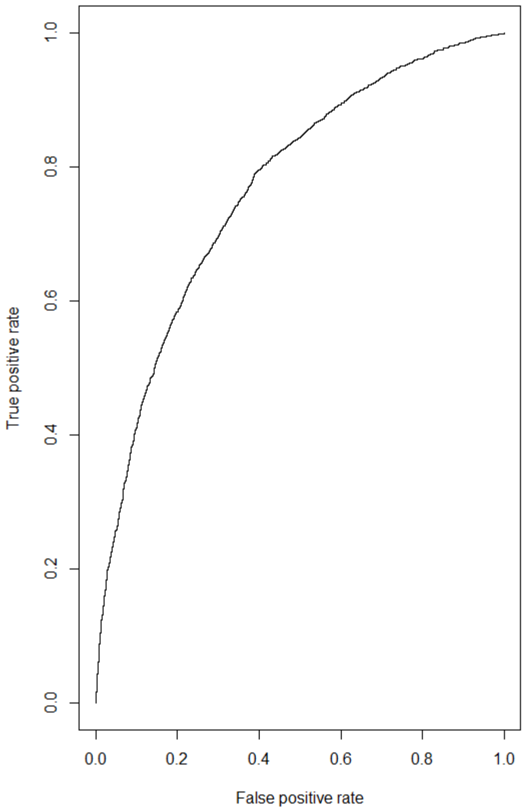
Assessing model using Confusion Matrix table in R :

```
> cm=table(y_pred,Churn_Modelling$Exited)
> cm

y_pred    0    1
     0 7681 1605
     1  282  432
> TN<-cm[1]
> TN
[1] 7681
> FN<-cm[2]
> FN
[1] 282
> FP<-cm[3]
> FP
[1] 1605
> TP<-cm[4]
> TP
[1] 432
> ActualYes<-FN+TP
> ActualYes
[1] 714
> ActualNo<-TN+FP
> ActualNo
[1] 9286
> PredictedYes<-FP+TP
> PredictedYes
[1] 2037
> PredictedNo<-TN+FN
> PredictedNo
[1] 7963
> accuracy1=mean(y_pred==Churn_Modelling$Exited)
> accuracy1
[1] 0.8113
> MR<-(FP+FN)/(TN+FN+FP+TP)
> MR
[1] 0.1887
> TPR<-TP/(ActualYes)
> TPR
[1] 0.605042
> FPR<-FP/(ActualNo)
> FPR
[1] 0.1728408
> precision1=precision<-diag(cm)/colSums(cm)
> precision1
        0         1
0.9645862 0.2120766
> recall1=recall<-diag(cm)/rowSums(cm)
> recall1
        0         1
0.8271592 0.6050420
>
```

```
y_pred    0    1
     0 7681 1605
     1  282  432
> myROC<-roc(response=Churn_Modelling$Exited,predictor = prob_predict1,positve='prob_predict1')
> plot(myROC)
> auc(roc(Churn_Modelling$Exited,prob_predict1))
Area under the curve: 0.7669
```



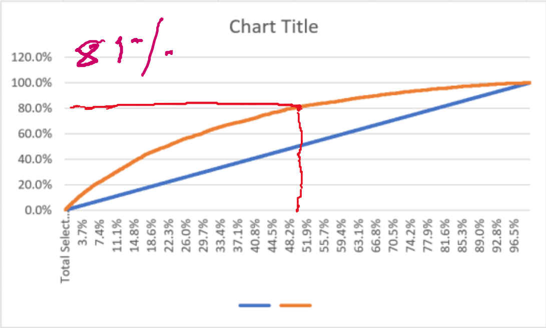

```
> roc.perf=performance(pred1,measure = "tpr",x.measure = "fpr")
> plot(roc.perf)
>
```



Assessing my model:

1. <u>Build a CAP curve</u>

	A	B	C	D	E	F	G	H	I
2	stats								
3	Total Exited	2037							
4	Total Records	10000							
5	Exit Ratio	20.4%							
13		Exited	P_hat	Total Select	Total Select Percent	Random Select	Random Select Percen	Model Sele	Model Select Percent
14		0	0.930669	1	0.0%	0.2037	0.0%	0	0.0%
15		1	0.928311	2	0.0%	0.4074	0.0%	1	0.0%
16		0	0.923303	3	0.0%	0.6111	0.0%	1	0.0%
17		1	0.906178	4	0.0%	0.8148	0.0%	2	0.1%
18		1	0.904457	5	0.1%	1.0185	0.1%	3	0.1%
19		1	0.892242	6	0.1%	1.2222	0.1%	4	0.2%
20		1	0.882859	7	0.1%	1.4259	0.1%	5	0.2%
21		0	0.876197	8	0.1%	1.6296	0.1%	5	0.2%
22		1	0.87429	9	0.1%	1.8333	0.1%	6	0.3%
23		1	0.871828	10	0.1%	2.037	0.1%	7	0.3%
24		1	0.861915	11	0.1%	2.2407	0.1%	8	0.4%
25		0	0.859745	12	0.1%	2.4444	0.1%	8	0.4%
26		1	0.857005	13	0.1%	2.6481	0.1%	9	0.4%
27		1	0.842062	14	0.1%	2.8518	0.1%	10	0.5%
28		1	0.839912	15	0.2%	3.0555	0.2%	11	0.5%
29		1	0.83736	16	0.2%	3.2592	0.2%	12	0.6%
30		0	0.834867	17	0.2%	3.4629	0.2%	12	0.6%
31		1	0.832188	18	0.2%	3.6666	0.2%	13	0.6%
32		0	0.830752	19	0.2%	3.8703	0.2%	13	0.6%
33		1	0.830146	20	0.2%	4.074	0.2%	14	0.7%
34		1	0.826519	21	0.2%	4.2777	0.2%	15	0.7%
35		0	0.821813	22	0.2%	4.4814	0.2%	15	0.7%
36		0	0.818944	23	0.2%	4.6851	0.2%	15	0.7%
37		1	0.818496	24	0.2%	4.8888	0.2%	16	0.8%
38		0	0.817436	25	0.3%	5.0925	0.3%	16	0.8%
39		0	0.816555	26	0.3%	5.2962	0.3%	16	0.8%



Churn-Modelling forecast.xlsx

Same 81% I showed in R studio by using accuracy = mean(y_pred==ChurnModelling$Exited)

So, this is all we did for the train data set, train model basically has accuracy of 81%. It's a good model, not best. But, we have to use test data to prove if the model behaves well with the new data set.

Test Data: Additional 1000 added and when model trying to predict ->75% which is less than 80%(trained model), since here only less data set, so more jagged lines.