Building a robust Geodemographic Segmentation Model

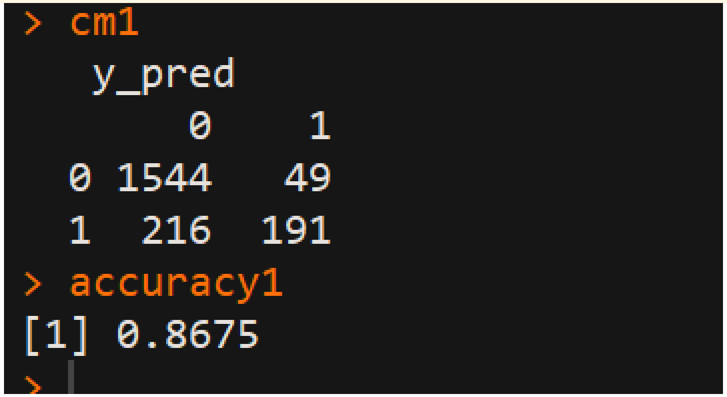
Applying Artificial Neural Network and step by step building a model

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

Used H2O package – deep learning package

```
#Artificial Neural Network

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

Churn_Modelling<-Churn_Modelling[4:14]

#install.packages("plyr")
library(plyr)
#encoding the categorical variables as factors
Churn_Modelling$Geography = as.numeric(factor(Churn_Modelling$Geography,levels =
c('France','Spain','Germany'),labels=c(1,2,3)))
Churn_Modelling$Gender = as.numeric(factor(Churn_Modelling$Gender,levels = c('Female','Male'),labels=c(1,2)))

#splitting the dataset into training and test sets, also install caTools packages
#install.packages('caTools')
library(caTools)
set.seed(123)
split=sample.split(Churn_Modelling$Exited,SplitRatio = 0.8)
training_set=subset(Churn_Modelling,split==TRUE)
test_set=subset(Churn_Modelling,split==FALSE)

#Feature Scaling or normailizing

training_set[-11]=scale(training_set[-11])
test_set[-11]=scale(test_set[-11])
```

```
#Fitting ANN to the training set

#install.packages("h2o")
library(h2o)
h2o.init(nthreads = -1)#building a deep learning model

classifier=h2o.deeplearning(y='Exited',training_frame = as.h2o(training_set),activation = 'Rectifier',hidden =
c(6,6),epochs = 100,train_samples_per_iteration = -2)

#Predicting the training results
prob_predictt=h2o.predict(classifier,newdata = as.h2o(test_set[-11]))#remove the dependent variable
y_pred=ifelse(prob_predictt>0.5,1,0)
y_pred<-as.vector(y_pred)
#making the confusion matrix

cm1=table(test_set$Exited,y_pred)
accuracy1=mean(y_pred==test_set$Exited)

#h20 shutdown
H2o.shutdown()
```



```
> cm1
   y_pred
       0    1
  0 1544   49
  1  216  191
> accuracy1
[1] 0.8675
> 
```