Bank Customer Churn

Jintawee.s

Review Data

```
myData <- read.csv(file = 'Churn Modeling.csv')
head(myData)
```

```
##   RowNumber CustomerId  Surname CreditScore Geography Gender Age Tenure
## 1         1   15634602 Hargrave         619    France Female  42      2
## 2         2   15647311     Hill         608     Spain Female  41      1
## 3         3   15619304     Onio         502    France Female  42      8
## 4         4   15701354     Boni         699    France Female  39      1
## 5         5   15737888 Mitchell         850     Spain Female  43      2
## 6         6   15574012      Chu         645     Spain   Male  44      8
##     Balance NumOfProducts HasCrCard IsActiveMember EstimatedSalary Exited
## 1      0.00             1         1              1       101348.88      1
## 2  83807.86             1         0              1       112542.58      0
## 3 159660.80             3         1              0       113931.57      1
## 4      0.00             2         0              0        93826.63      0
## 5 125510.82             1         1              1        79084.10      0
## 6 113755.78             2         1              0       149756.71      1
```

Drop NA (missing values)

```
myData <- na.omit(myData)
nrow(myData)
```

```
## [1] 10000
```

Convert gender to factor

```
myData$Gender = as.factor(myData$Gender)
str(myData)
```

```
## 'data.frame':    10000 obs. of  14 variables:
##  $ RowNumber      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ CustomerId     : int  15634602 15647311 15619304 15701354 15737888 15574012 15592531 15656148 157
##  $ Surname        : chr  "Hargrave" "Hill" "Onio" "Boni" ...
##  $ CreditScore    : int  619 608 502 699 850 645 822 376 501 684 ...
##  $ Geography      : chr  "France" "Spain" "France" "France" ...
##  $ Gender         : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 2 2 1 2 2 ...
##  $ Age            : int  42 41 42 39 43 44 50 29 44 27 ...
##  $ Tenure         : int  2 1 8 1 2 8 7 4 4 2 ...
##  $ Balance        : num  0 83808 159661 0 125511 ...
```

```
##  $ NumOfProducts  : int  1 1 3 2 1 2 2 4 2 1 ...
##  $ HasCrCard      : int  1 0 1 0 1 1 1 1 0 1 ...
##  $ IsActiveMember : int  1 1 0 0 1 0 1 0 1 1 ...
##  $ EstimatedSalary: num  101349 112543 113932 93827 79084 ...
##  $ Exited         : int  1 0 1 0 0 1 0 1 0 0 ...
```

Split Data

```
set.seed (59)
n <- nrow(myData)
id <- sample(1:n, size=n*0.7)
train_data <- myData[id, ]
test_data <- myData[-id, ]
```

Train Model

```
model_train <- glm(Exited ~ CreditScore + Gender + Age + Balance + IsActiveMember,
               data = train_data, family="binomial")
summary(model_train)
```

```
##
## Call:
## glm(formula = Exited ~ CreditScore + Gender + Age + Balance +
##     IsActiveMember, family = "binomial", data = train_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1358  -0.6739  -0.4730  -0.2810   2.9238
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.397e+00  2.540e-01 -13.375   <2e-16 ***
## CreditScore    -8.388e-04  3.308e-04  -2.536   0.0112 *
## GenderMale     -5.975e-01  6.407e-02  -9.326   <2e-16 ***
## Age             7.112e-02  2.994e-03  23.753   <2e-16 ***
## Balance         5.504e-06  5.284e-07  10.417   <2e-16 ***
## IsActiveMember -1.006e+00  6.723e-02 -14.959   <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: 7158.1  on 6999  degrees of freedom
## Residual deviance: 6170.3  on 6994  degrees of freedom
## AIC: 6182.3
##
## Number of Fisher Scoring iterations: 5
```

Predict and Evaluate Model

```
train_data$prob_Exited <- predict(model_train, type="response")
train_data$preb_Exited <- ifelse(train_data$prob_Exited >=0.5, 1, 0)
```

Confusion matrix

```
conM_train <- table(train_data$preb_Exited, train_data$Exited,
              dnn=c("Predicted", "Actual"))
```

Model_train Evaluation

```
Acc_train <- (conM_train[1 , 1] + conM_train[2 , 2]) / sum(conM_train)
Pre_train <- conM_train[2 ,2]/ (conM_train[2 ,1] + conM_train[2 ,2])
Re_train <- conM_train[2 ,2]/ (conM_train[1 ,2] + conM_train[2 ,2])

F1_train <- 2*((Pre_train*Re_train) / (Pre_train+Re_train))

cat("Accuracy:", Acc_train, "\nPrecision:", Pre_train, "\nRecall:", Re_train, "\nF1:", F1_train)
```

```
## Accuracy: 0.8002857
## Precision: 0.5647321
## Recall: 0.1737637
## F1: 0.2657563
```

Test Model

```
model_test <- glm(Exited ~ CreditScore + Gender + Age + Balance + IsActiveMember,
              data = test_data, family="binomial")
summary(model_test)
```

```
##
## Call:
## glm(formula = Exited ~ CreditScore + Gender + Age + Balance +
##     IsActiveMember, family = "binomial", data = test_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8921  -0.6647  -0.4408  -0.2697   2.9915
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -4.046e+00  3.964e-01 -10.207  < 2e-16 ***
## CreditScore    -1.985e-04  5.117e-04  -0.388    0.698
## GenderMale     -4.107e-01  1.002e-01  -4.097 4.18e-05 ***
## Age             7.739e-02  4.877e-03  15.867  < 2e-16 ***
## Balance         4.251e-06  8.231e-07   5.165 2.40e-07 ***
## IsActiveMember -1.266e+00  1.095e-01 -11.565  < 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: 2949.0  on 2999  degrees of freedom
## Residual deviance: 2528.8  on 2994  degrees of freedom
## AIC: 2540.8
##
## Number of Fisher Scoring iterations: 5
```

Predict and Evaluate Model

```
test_data$prob_Exited <- predict(model_test, type="response")
test_data$preb_Exited <- ifelse(test_data$prob_Exited >=0.5, 1, 0)
```

Confusion matrix

```
conM_test <- table(test_data$preb_Exited, test_data$Exited,
              dnn=c("Predicted", "Actual"))
```

Model_test Evaluation

```
Acc_test <- (conM_test[1 , 1] + conM_test[2 , 2]) / sum(conM_test)
Pre_test <- conM_test[2 ,2]/ (conM_test[2 ,1] + conM_test[2 ,2])
Re_test <- conM_test[2 ,2]/ (conM_test[1 ,2] + conM_test[2 ,2])

F1_test <- 2*((Pre_test*Re_test) / (Pre_test+Re_test))

cat("Accuracy:", Acc_test, "\nPrecision:", Pre_test, "\nRecall:", Re_test, "\nF1:", F1_test)
```

```
## Accuracy: 0.8176667
## Precision: 0.6103896
## Recall: 0.16179
## F1: 0.2557823
```