ChurnPrediction

Kushal Agarwal

2024-03-18

```
churn_data = read.csv('/Users/kushalagarwal/Downloads/customer_churn.csv')
# Look at the first 6 observations
head(churn_data)
```

```
##   Call.Failure Complaints Subscription.Length Charge.Amount Seconds.of.Use
## 1            8          0                  38             0           4370
## 2            0          0                  39             0            318
## 3           10          0                  37             0           2453
## 4           10          0                  38             0           4198
## 5            3          0                  38             0           2393
## 6           11          0                  38             1           3775
##   Frequency.of.use Frequency.of.SMS Distinct.Called.Numbers Age.Group
## 1               71                5                      17         3
## 2                5                7                       4         2
## 3               60              359                      24         3
## 4               66                1                      35         1
## 5               58                2                      33         1
## 6               82               32                      28         3
##   Tariff.Plan Status Age Customer.Value Churn
## 1           1      1  30        197.640     0
## 2           1      2  25         46.035     0
## 3           1      1  30       1536.520     0
## 4           1      1  15        240.020     0
## 5           1      1  15        145.805     0
## 6           1      1  30        282.280     0
```

```
# Check the dimension
dim(churn_data)
```

```
## [1] 3150   14
```

```
# Change the column names
names(churn_data) = gsub(" ", "", names(churn_data))
head(churn_data)
```

```
##   Call.Failure Complaints Subscription.Length Charge.Amount Seconds.of.Use
## 1            8          0                  38             0           4370
## 2            0          0                  39             0            318
## 3           10          0                  37             0           2453
## 4           10          0                  38             0           4198
```

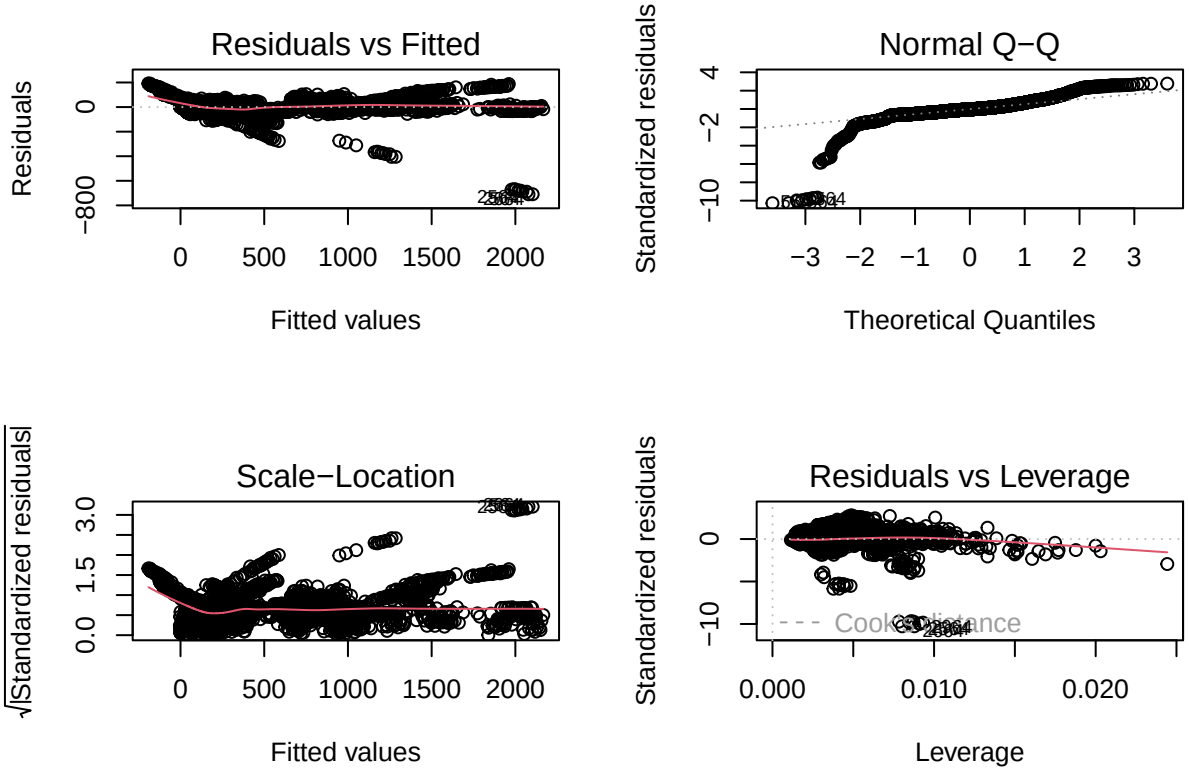
```
## 5            3          0                  38             0           2393
## 6           11          0                  38             1           3775
##   Frequency.of.use Frequency.of.SMS Distinct.Called.Numbers Age.Group
## 1               71                5                      17         3
## 2                5                7                       4         2
## 3               60              359                      24         3
## 4               66                1                      35         1
## 5               58                2                      33         1
## 6               82               32                      28         3
##   Tariff.Plan Status Age Customer.Value Churn
## 1           1      1  30        197.640     0
## 2           1      2  25         46.035     0
## 3           1      1  30       1536.520     0
## 4           1      1  15        240.020     0
## 5           1      1  15        145.805     0
## 6           1      1  30        282.280     0
```

```
# Fit the multiple linear regression model
cust_value_model = lm(formula = Customer.Value ~ Call.Failure +
                        Complaints + Subscription.Length + Charge.Amount +
                        Seconds.of.Use +Frequency.of.use + Frequency.of.SMS +
                        Distinct.Called.Numbers + Age.Group + Tariff.Plan +
                        Status + Age,data = churn_data)
summary(cust_value_model)
```

```
##
## Call:
## lm(formula = Customer.Value ~ Call.Failure + Complaints + Subscription.Length +
##     Charge.Amount + Seconds.of.Use + Frequency.of.use + Frequency.of.SMS +
##     Distinct.Called.Numbers + Age.Group + Tariff.Plan + Status +
##     Age, data = churn_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -709.81  -26.48   -2.63   24.24  191.43
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             160.201207  10.656561  15.033  < 2e-16 ***
## Call.Failure             -0.489519   0.290861  -1.683 0.092474 .
## Complaints                7.227189   5.000052   1.445 0.148439
## Subscription.Length       0.741287   0.156189   4.746 2.17e-06 ***
## Charge.Amount           -14.298831   1.428753 -10.008  < 2e-16 ***
## Seconds.of.Use            0.047845   0.001116  42.875  < 2e-16 ***
## Frequency.of.use         -0.540230   0.093055  -5.805 7.06e-09 ***
## Frequency.of.SMS          4.010644   0.012108 331.234  < 2e-16 ***
## Distinct.Called.Numbers   0.363675   0.112751   3.225 0.001271 **
## Age.Group                -7.254712   5.211649  -1.392 0.164015
## Tariff.Plan              75.507316   5.669970  13.317  < 2e-16 ***
## Status                  -12.824538   3.884456  -3.302 0.000972 ***
## Age                      -7.098635   0.524340 -13.538  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 69.36 on 3137 degrees of freedom
## Multiple R-squared:  0.9821, Adjusted R-squared:  0.982
## F-statistic: 1.432e+04 on 12 and 3137 DF,  p-value: < 2.2e-16
```

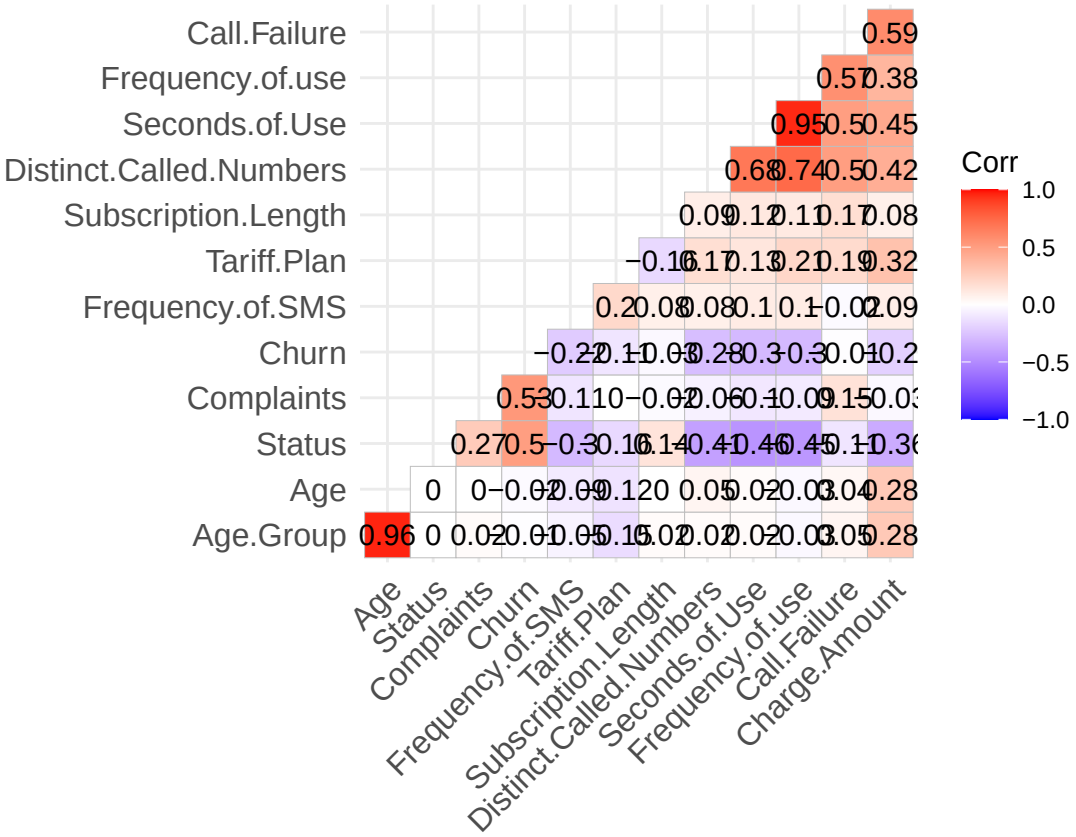
```
par(mfrow= c(2,2))
plot(cust_value_model)
```




```
library(ggcorrplot)
```

```
## Loading required package: ggplot2
```

```
# Remove the Customer Value column
reduced_data <- subset(churn_data, select = -Customer.Value)

# Compute correlation at 2 decimal places
corr_matrix = round(cor(reduced_data), 2)

# Compute and show the  result
ggcorrplot(corr_matrix, hc.order = TRUE, type = "lower",
           lab = TRUE)
```




```
# removing predictors with high multi colinearity
# (age group & seconds of use)
colinearity_model = lm(formula = Customer.Value ~ Call.Failure +
                         Complaints + Subscription.Length + Charge.Amount +
                         Seconds.of.Use + Frequency.of.SMS +
                         Distinct.Called.Numbers + Tariff.Plan +
                         Status + Age,data = churn_data)
summary(colinearity_model)
```

```
##
## Call:
## lm(formula = Customer.Value ~ Call.Failure + Complaints + Subscription.Length +
##     Charge.Amount + Seconds.of.Use + Frequency.of.SMS + Distinct.Called.Numbers +
##     Tariff.Plan + Status + Age, data = churn_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -705.70  -23.60   -3.71   23.56  192.33
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.653e+02  1.068e+01  15.475  < 2e-16 ***
## Call.Failure            -1.377e+00  2.506e-01  -5.494 4.25e-08 ***
## Complaints               9.232e+00  4.997e+00   1.848   0.0648 .
## Subscription.Length      7.313e-01  1.570e-01   4.659 3.31e-06 ***
## Charge.Amount           -1.008e+01  1.217e+00  -8.288  < 2e-16 ***
```

```
## Seconds.of.Use           4.186e-02  4.427e-04  94.540  < 2e-16 ***
## Frequency.of.SMS         4.010e+00  1.201e-02 333.854  < 2e-16 ***
## Distinct.Called.Numbers  1.461e-01  1.041e-01   1.403   0.1606
## Tariff.Plan              6.532e+01  5.257e+00  12.424  < 2e-16 ***
## Status                  -8.420e+00  3.830e+00  -2.199   0.0280 *
## Age                     -7.832e+00  1.548e-01 -50.578  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 69.74 on 3139 degrees of freedom
## Multiple R-squared:  0.9819, Adjusted R-squared:  0.9818
## F-statistic: 1.699e+04 on 10 and 3139 DF,  p-value: < 2.2e-16
```

```
# Anova test
anova(cust_value_model, colinearity_model)
```

```
## Analysis of Variance Table
##
## Model 1: Customer.Value ~ Call.Failure + Complaints + Subscription.Length +
##     Charge.Amount + Seconds.of.Use + Frequency.of.use + Frequency.of.SMS +
##     Distinct.Called.Numbers + Age.Group + Tariff.Plan + Status +
##     Age
## Model 2: Customer.Value ~ Call.Failure + Complaints + Subscription.Length +
##     Charge.Amount + Seconds.of.Use + Frequency.of.SMS + Distinct.Called.Numbers +
##     Tariff.Plan + Status + Age
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1   3137 15092825
## 2   3139 15268753 -2   -175929 18.283 1.275e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# removing predictors that do not habve a high individual statistical signficance
significant_only_model = lm(formula = Customer.Value ~
                              Subscription.Length + Charge.Amount +
                              Seconds.of.Use +Frequency.of.use + Frequency.of.SMS +
                              Distinct.Called.Numbers + Tariff.Plan +
                              Status + Age,data = churn_data)
summary(significant_only_model)
```

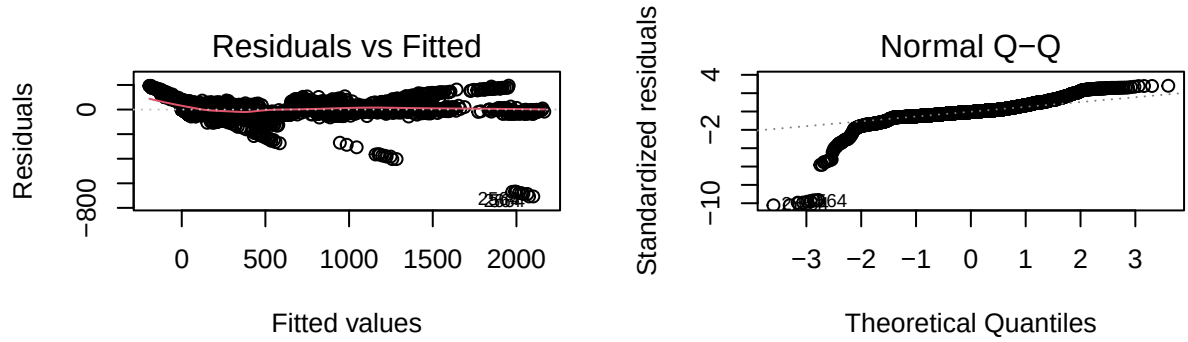
```
##
## Call:
## lm(formula = Customer.Value ~ Subscription.Length + Charge.Amount +
##     Seconds.of.Use + Frequency.of.use + Frequency.of.SMS + Distinct.Called.Numbers +
##     Tariff.Plan + Status + Age, data = churn_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -706.92  -25.94   -4.10   23.20  194.26
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             159.600223  10.646945  14.990  < 2e-16 ***
## Subscription.Length       0.706071   0.155474   4.541 5.80e-06 ***
```

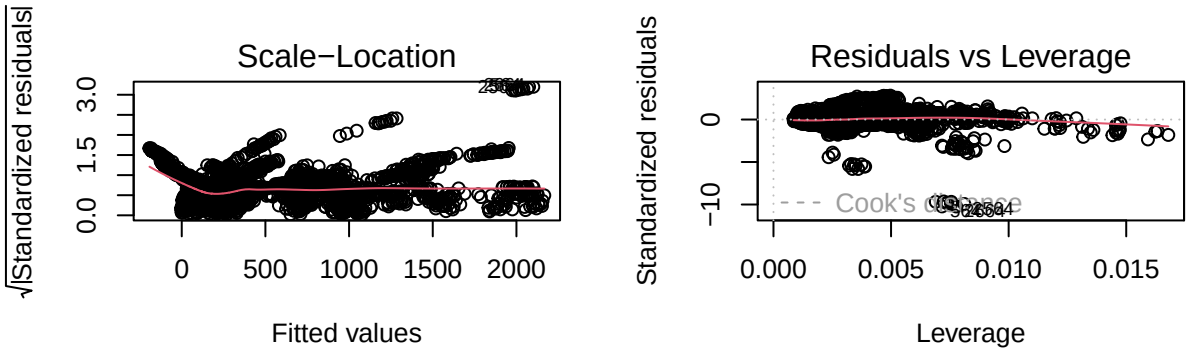
```
## Charge.Amount           -15.935690   1.103039 -14.447  < 2e-16 ***
## Seconds.of.Use            0.048607   0.001022  47.543  < 2e-16 ***
## Frequency.of.use         -0.625734   0.079631  -7.858 5.32e-15 ***
## Frequency.of.SMS          4.008469   0.011945 335.584  < 2e-16 ***
## Distinct.Called.Numbers   0.390373   0.111383   3.505 0.000463 ***
## Tariff.Plan              78.641296   5.490352  14.324  < 2e-16 ***
## Status                  -13.845853   3.594606  -3.852 0.000120 ***
## Age                      -7.757874   0.152254 -50.954  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 69.39 on 3140 degrees of freedom
## Multiple R-squared:  0.982,  Adjusted R-squared:  0.982
## F-statistic: 1.907e+04 on 9 and 3140 DF,  p-value: < 2.2e-16
```

```
# F-statistic: 1.907e+04 on 9 and 3140 DF,  p-value: < 2.2e-16

anova(cust_value_model, significant_only_model)
```

```
## Analysis of Variance Table
##
## Model 1: Customer.Value ~ Call.Failure + Complaints + Subscription.Length +
##     Charge.Amount + Seconds.of.Use + Frequency.of.use + Frequency.of.SMS +
##     Distinct.Called.Numbers + Age.Group + Tariff.Plan + Status +
##     Age
## Model 2: Customer.Value ~ Subscription.Length + Charge.Amount + Seconds.of.Use +
##     Frequency.of.use + Frequency.of.SMS + Distinct.Called.Numbers +
##     Tariff.Plan + Status + Age
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1   3137 15092825
## 2   3140 15120908 -3    -28083 1.9457 0.1201
```

```
# 0.1201, so reduced better

par(mfrow= c(2,2))
plot(significant_only_model)
```






```
#### prediction

# Convert Churn to a factor
churn_data$Churn <- as.factor(churn_data$Churn)

# Splitting the dataset into training and testing sets
set.seed(123) # For reproducibility
train_indices <- sample(1:nrow(churn_data), size = 0.8*nrow(churn_data))
train_data <- churn_data[train_indices, ]
test_data <- churn_data[-train_indices, ]

# Fitting a logistic regression model
churn_model <- glm(Churn ~ Call.Failure + Complaints + Subscription.Length + Charge.Amount +
                     Seconds.of.Use + Frequency.of.use + Frequency.of.SMS +
                     Distinct.Called.Numbers + Age.Group + Tariff.Plan +
                     Status + Age,
                   data = train_data, family = binomial)

# Summary of the model to check coefficients and overall fit
summary(churn_model)
```

```
##
## Call:
## glm(formula = Churn ~ Call.Failure + Complaints + Subscription.Length +
##     Charge.Amount + Seconds.of.Use + Frequency.of.use + Frequency.of.SMS +
##     Distinct.Called.Numbers + Age.Group + Tariff.Plan + Status +
```

```
##     Age, family = binomial, data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.62275  -0.32993  -0.13660  -0.04344   3.06619
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -2.0232832  0.8270463  -2.446 0.014429 *
## Call.Failure              0.1379478  0.0193041   7.146 8.93e-13 ***
## Complaints                4.0122518  0.3210618  12.497  < 2e-16 ***
## Subscription.Length      -0.0313341  0.0106585  -2.940 0.003284 **
## Charge.Amount            -0.5988464  0.1304745  -4.590 4.44e-06 ***
## Seconds.of.Use            0.0004045  0.0001152   3.512 0.000445 ***
## Frequency.of.use         -0.0517195  0.0089101  -5.805 6.45e-09 ***
## Frequency.of.SMS         -0.0119288  0.0027485  -4.340 1.42e-05 ***
## Distinct.Called.Numbers -0.0155975  0.0105028  -1.485 0.137524
## Age.Group                 0.3347906  0.3199694   1.046 0.295413
## Tariff.Plan               0.6807215  0.6862236   0.992 0.321206
## Status                    1.2090526  0.2158641   5.601 2.13e-08 ***
## Age                      -0.0422105  0.0328786  -1.284 0.199201
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2168.2  on 2519  degrees of freedom
## Residual deviance: 1111.6  on 2507  degrees of freedom
## AIC: 1137.6
##
## Number of Fisher Scoring iterations: 8
```

```
# Predicting on the test set
predictions <- predict(churn_model, test_data, type = "response")
predicted_classes <- ifelse(predictions > 0.5, 1, 0)

# Evaluating the model
library(caret)
```

```
## Loading required package: lattice
```

```
confusionMatrix(factor(predicted_classes), test_data$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 511  54
##          1  13  52
##
##                Accuracy : 0.8937
##                  95% CI : (0.8669, 0.9166)
##     No Information Rate : 0.8317
```

```
##     P-Value [Acc > NIR] : 7.565e-06
##
##                   Kappa : 0.5507
##
##  Mcnemar's Test P-Value : 1.025e-06
##
##             Sensitivity : 0.9752
##             Specificity : 0.4906
##          Pos Pred Value : 0.9044
##          Neg Pred Value : 0.8000
##              Prevalence : 0.8317
##          Detection Rate : 0.8111
##    Detection Prevalence : 0.8968
##       Balanced Accuracy : 0.7329
##
##        'Positive' Class : 0
##
```