In the cellphone churn data, we can see that the total number of records is 3333. Out of the 3333, Churn=1 is 483.

So, 483 \* 100/3333 = 14.49% data is having churn=1 and rest 85.5% are having churn=0.

The data is highly skewed, as the dataset has very less amount data having churn=1

> table(cellphone$Churn)

0 1

2850 483

set.seed(250)

> indexes=sample(1:nrow(cell),size = (nrow(cell)\*0.7))

devcell <- Cellphone[indexes,]

testcell <- Cellphone[-indexes,]

From the sample data we have created, development sample of 70% and test sample of 30% of the total data. From the below figure, 14.31% of the dev data has churn=1 and rest 85.69% data has Churn=0

> table(devcell$Churn)

0 1

1999 334

In the test sample we can see that 14.9% of test data has Churn=1 and rest 85.1% has Churn=0

> table(testcell$Churn)

0 1

851 149

As per the problem statement, We need to build a model to predict the probability for a customer to churn`

. As the Churn is dichotomous binomial in nature, so we can’t apply Linear Regression model. It can produce the probability more than 1 and less than 0.

So we need to build a logistic regression as the predicted variable is not a continuous one.

let’s create the logistic model with taking 2 way interaction as well. With the stepwise glm, we can compare the AIC to find the best combination of independent parameters. The combination with the lowest AIC will be the best choice.

> glm1<- glm(Churn ~ .,data=devcell, family = binomial)

> search <- step(glm1,scope = .~.^2,direction = 'forward', trace = 0, k =2)

From the above steps, we have pulled the best logistic model with the lowest AIC number.

Step: AIC= 1178

Call:

glm(formula = Churn ~ AccountWeeks + ContractRenewal + DataPlan +

DataUsage + CustServCalls + DayMins + DayCalls + MonthlyCharge +

OverageFee + RoamMins + CustServCalls:DayMins + DataPlan:MonthlyCharge +

DayMins:MonthlyCharge + ContractRenewal:RoamMins + CustServCalls:OverageFee +

ContractRenewal:DayMins + DataUsage:DayMins + DayCalls:OverageFee +

ContractRenewal:CustServCalls + DataUsage:RoamMins + DayMins:OverageFee +

DataUsage:CustServCalls + ContractRenewal:DataPlan + ContractRenewal:MonthlyCharge +

ContractRenewal:DayCalls + DataPlan:OverageFee + DataUsage:OverageFee +

AccountWeeks:DataPlan, family = binomial, data = devcell)

With the above combination of independent variables we will build the logistic model.

> glm2 <- glm(formula = Churn ~ AccountWeeks + ContractRenewal + DataPlan +

+ DataUsage + CustServCalls + DayMins + DayCalls + MonthlyCharge +

+ OverageFee + RoamMins + CustServCalls:DayMins + DataPlan:MonthlyCharge +

+ DayMins:MonthlyCharge + ContractRenewal:RoamMins + CustServCalls:OverageFee +

+ ContractRenewal:DayMins + DataUsage:DayMins + DayCalls:OverageFee +

+ ContractRenewal:CustServCalls + DataUsage:RoamMins + DayMins:OverageFee +

+ DataUsage:CustServCalls + ContractRenewal:DataPlan + ContractRenewal:MonthlyCharge +

+ ContractRenewal:DayCalls + DataPlan:OverageFee + DataUsage:OverageFee +

+ AccountWeeks:DataPlan, family = binomial, data = devcell)

>

> summary(glm2)

Call:

glm(formula = Churn ~ AccountWeeks + ContractRenewal + DataPlan +

DataUsage + CustServCalls + DayMins + DayCalls + MonthlyCharge +

OverageFee + RoamMins + CustServCalls:DayMins + DataPlan:MonthlyCharge +

DayMins:MonthlyCharge + ContractRenewal:RoamMins + CustServCalls:OverageFee +

ContractRenewal:DayMins + DataUsage:DayMins + DayCalls:OverageFee +

ContractRenewal:CustServCalls + DataUsage:RoamMins + DayMins:OverageFee +

DataUsage:CustServCalls + ContractRenewal:DataPlan + ContractRenewal:MonthlyCharge +

ContractRenewal:DayCalls + DataPlan:OverageFee + DataUsage:OverageFee +

AccountWeeks:DataPlan, family = binomial, data = devcell)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0833 -0.4023 -0.2101 -0.0923 3.7017

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 3.7542838 2.5535470 1.470 0.141501

AccountWeeks 0.0031253 0.0021330 1.465 0.142868

ContractRenewal1 -5.3649175 1.5111473 -3.550 0.000385 \*\*\*

DataPlan1 14.4324176 3.2494900 4.441 8.94e-06 \*\*\*

DataUsage -2.8226219 2.9661576 -0.952 0.341295

CustServCalls 3.0833962 0.3487946 8.840 < 2e-16 \*\*\*

DayMins -0.0358492 0.0475532 -0.754 0.450924

DayCalls -0.0648899 0.0175335 -3.701 0.000215 \*\*\*

MonthlyCharge -0.1643170 0.2740338 -0.600 0.548757

OverageFee -0.3973588 0.5133479 -0.774 0.438899

RoamMins 0.4766349 0.0804326 5.926 3.11e-09 \*\*\*

CustServCalls:DayMins -0.0095431 0.0010651 -8.960 < 2e-16 \*\*\*

DataPlan1:MonthlyCharge -0.0240817 0.0437235 -0.551 0.581791

DayMins:MonthlyCharge 0.0008977 0.0001081 8.304 < 2e-16 \*\*\*

ContractRenewal1:RoamMins -0.4769375 0.0849255 -5.616 1.95e-08 \*\*\*

CustServCalls:OverageFee -0.1202287 0.0224336 -5.359 8.35e-08 \*\*\*

ContractRenewal1:DayMins 0.0022781 0.0072002 0.316 0.751702

DataUsage:DayMins -0.0139918 0.0030019 -4.661 3.15e-06 \*\*\*

DayCalls:OverageFee 0.0049176 0.0015509 3.171 0.001520 \*\*

ContractRenewal1:CustServCalls 0.6047596 0.1418130 4.264 2.00e-05 \*\*\*

DataUsage:RoamMins 0.1263478 0.0389007 3.248 0.001162 \*\*

DayMins:OverageFee 0.0010957 0.0005907 1.855 0.063627 .

DataUsage:CustServCalls 0.1204665 0.0433671 2.778 0.005472 \*\*

ContractRenewal1:DataPlan1 -4.0090034 1.1912621 -3.365 0.000764 \*\*\*

ContractRenewal1:MonthlyCharge 0.0929386 0.0364365 2.551 0.010751 \*

ContractRenewal1:DayCalls 0.0207747 0.0087257 2.381 0.017271 \*

DataPlan1:OverageFee -0.8652488 0.3181870 -2.719 0.006542 \*\*

DataUsage:OverageFee 0.2150736 0.1012739 2.124 0.033697 \*

AccountWeeks:DataPlan1 0.0083500 0.0053758 1.553 0.120365

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1916.2 on 2332 degrees of freedom

Residual deviance: 1119.7 on 2304 degrees of freedom

AIC: 1177.7

Number of Fisher Scoring iterations: 6

**HYPOTHESIS**.

As per the problem statement Churn is a function of (Accountweek, ContractRenewal, DataPlan, DataUsage, CustServCalls,DayMins, DayCalls, MonthlyCharge, OverageFee, RoamMins)

**Null Hypothesis:** All the betas are zero corresponding to the independent variables.

**Alternate Hypothesis:** At least one of the beta is not 0.

**Finding the Overall validity of the model:**

**LOG LIKELIHOOD**

1. From the log likelihood, we can see that, intercept only model -958.08 variance was unknown to me. In the intercept only model we are assuming that the Churn is not a function of any of the independent variable.
2. When take the full model, -559.86 variance was unknown to me. So we can say that,

1 – (-559.86 /-958.08)= 41.56% of the uncertainty inherent in the intercept only model is calibrated by the full model.

1. Chisq likelihood ratio is significant. So we can accept the Alternate Hypothesis that at least one of the beta is not zero. So Model is significant.

> library(lmtest)

> library(pscl)

> library(Deducer)

> lrtest(glm2)

Model 2: Churn ~ 1

#Df LogLik Df Chisq Pr(>Chisq)

1 29 -559.86

2 1 -958.08 -28 796.45 < 2.2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

**PSEUDO R SQUARE**

> pR2(glm2)

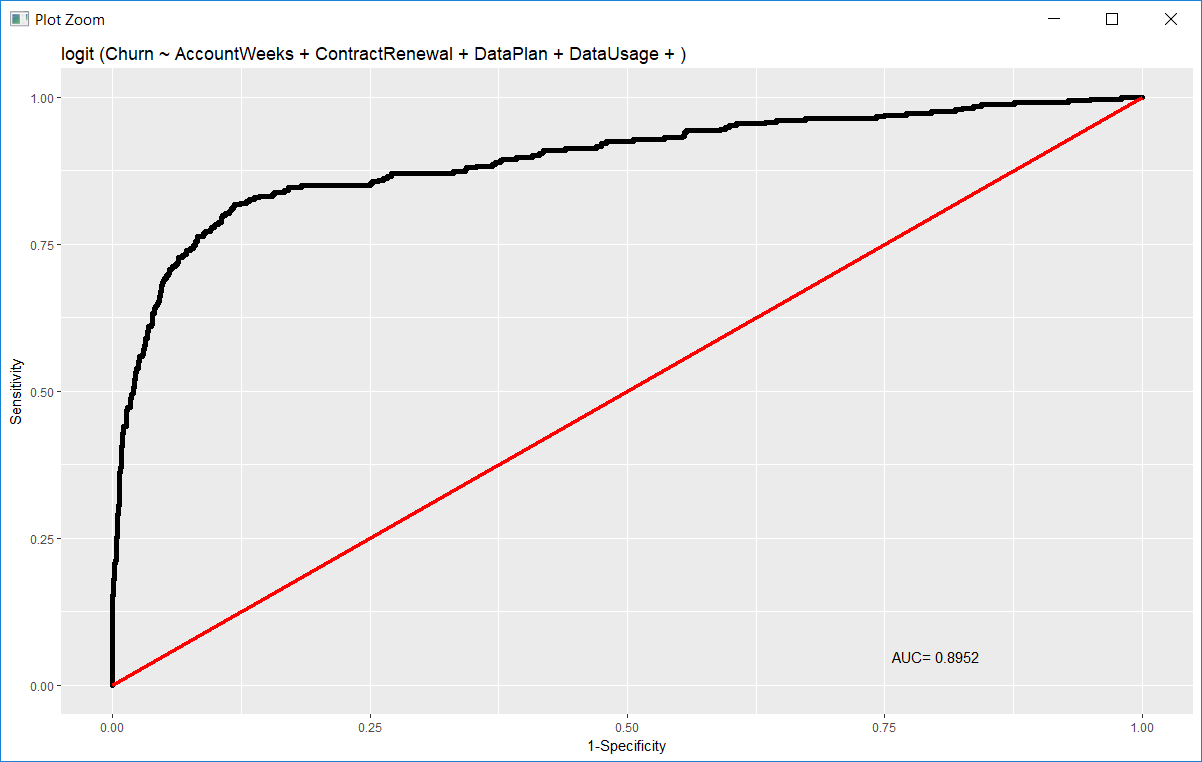
llh llhNull G2 McFadden r2ML r2CU

-559.8553539 -958.0803063 796.4499049 0.4156488 0.2892144 0.5163129

1. From the McFadden R square we can conclude that 41.56% uncertainty produced by the intercept model is explained by the full model. McFadden R square is a conservative one. Minimum 10% of McFadden R square should be there for validity of the model. In our case this is 41.56%.
2. So we can conclude that the model is significant. And We can accept the alternate hypothesis.

**ROC**

> rocplot(glm2)



**ACCURACY OF THE MODEL:**

Predicting the DEV SAMPLE.

> predict.dev <- predict(glm2,type='response')

> View(predict.dev)

> table(devcell$Churn,predict.dev > 0.5)

FALSE TRUE

0 1954 45

1 158 176

> library(ROSE)

> accuracy.meas(devcell$Churn,predict.dev)

Call:

accuracy.meas(response = devcell$Churn, predicted = predict.dev)

Examples are labelled as positive when predicted is greater than 0.5

precision: 0.796

recall: 0.527

F: 0.317

Predicting the Test Sample

> ## predicting the test sample

> predict.test <- predict(glm2,newdata = testcell, type = "response")

> accuracy.meas(testcell$Churn,predict.test)

Call:

accuracy.meas(response = testcell$Churn, predicted = predict.test)

Examples are labelled as positive when predicted is greater than 0.5

precision: 0.774

recall: 0.483

F: 0.298

We can see that, the accuracy is pretty much same in both the cases of dev and test sample data. But the model can not predict well for the customers with CHURN=1. In both the cases SENSITIVITY is around 48.3-52.7 % only.

To increase the sensitivity of the model we can do oversampling to balance the data.

> library(ROSE)

> cell.rose <- ROSE(Churn ~ AccountWeeks + ContractRenewal + DataPlan +

+ DataUsage + CustServCalls + DayMins + DayCalls + MonthlyCharge +

+ OverageFee + RoamMins + CustServCalls:DayMins + DataPlan:MonthlyCharge +

+ DayMins:MonthlyCharge + ContractRenewal:RoamMins + CustServCalls:OverageFee +

+ ContractRenewal:DayMins + DataUsage:DayMins + DayCalls:OverageFee +

+ ContractRenewal:CustServCalls + DataUsage:RoamMins + DayMins:OverageFee +

+ DataUsage:CustServCalls + ContractRenewal:DataPlan + ContractRenewal:MonthlyCharge +

+ ContractRenewal:DayCalls + DataPlan:OverageFee + DataUsage:OverageFee +

+ AccountWeeks:DataPlan

+ , data = devcell, seed = 1)$data

> table(cell.rose$Churn)

0 1

1217 1116

Now we can build the logistic model again on the ROSE data, and test that on the test sample and check the accuracy and sensitivity of the model.

> glm.rose <- glm(formula = Churn ~ AccountWeeks + ContractRenewal + DataPlan +

+ DataUsage + CustServCalls + DayMins + DayCalls + MonthlyCharge +

+ OverageFee + RoamMins + CustServCalls:DayMins + DataPlan:MonthlyCharge +

+ DayMins:MonthlyCharge + ContractRenewal:RoamMins + CustServCalls:OverageFee +

+ ContractRenewal:DayMins + DataUsage:DayMins + DayCalls:OverageFee +

+ ContractRenewal:CustServCalls + DataUsage:RoamMins + DayMins:OverageFee +

+ DataUsage:CustServCalls + ContractRenewal:DataPlan + ContractRenewal:MonthlyCharge +

+ ContractRenewal:DayCalls + DataPlan:OverageFee + DataUsage:OverageFee +

+ AccountWeeks:DataPlan,family = binomial,data=cell.rose)

Predicting the dev sample with the new model

> predict.devrose <- predict(glm.rose,type = 'response')

> table(cell.rose$Churn,predict.devrose > 0.5)

FALSE TRUE

0 1016 201

1 247 869

> ## measuring the accuracy and sensitivity

> accuracy.meas(cell.rose$Churn,predict.devrose)

Call:

accuracy.meas(response = cell.rose$Churn, predicted = predict.devrose)

Examples are labelled as positive when predicted is greater than 0.5

precision: 0.812

recall: 0.779

F: 0.398

Predicting the test sample with the new model.

> predict.test1 <- predict(glm.rose,newdata = testcell, type="response")

> accuracy.meas(testcell$Churn,predict.test1)

Call:

accuracy.meas(response = testcell$Churn, predicted = predict.test1)

Examples are labelled as positive when predicted is greater than 0.5

precision: 0.494

recall: 0.859

F: 0.314

> table(testcell$Churn,predict.test1 > 0.5)

FALSE TRUE

0 720 131

1 21 128

From the above result we can see that the after doing the oversampling, the model is pretty good in predicting the churn = 1 and Sensitivity of the model has increased a lot. But at the same time we can see that the overall precision has decreased. The model is failing to identify those customers who will not be churning.

We can play around with the cutoff to do a balance in the precision and recall of the model.

**Finding the importance of the dependent variable.**

To find the importance let’s see the coefficient and odds of the best model.

> odds.rose$coef <- as.data.frame( coef(glm.rose))

> odds.rose$odds <- as.data.frame(exp(coef(glm.rose)))

> odds.rose

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Coefficient** | **ODDS** | **REL. Importance** |
| DataPlan1 | 5.755622 | 315.961900 | 91.43638693 |
| CustServCalls | 1.601470 | 4.960319 | 1.435469427 |
| ContractRenewal1:CustServCalls | 0.391321 | 1.478933 | 0.427989229 |
| RoamMins | 0.250996 | 1.285305 | 0.371955116 |
| DataUsage | 0.118118 | 1.125377 | 0.325673465 |
| DataUsage:CustServCalls | 0.092034 | 1.096402 | 0.317288374 |
| DataUsage:RoamMins | 0.026224 | 1.026571 | 0.297079943 |
| DataUsage:OverageFee | 0.018438 | 1.018609 | 0.294775815 |
| ContractRenewal1:DayCalls | 0.017310 | 1.017460 | 0.294443306 |
| ContractRenewal1:MonthlyCharge | 0.015211 | 1.015328 | 0.293826325 |
| ContractRenewal1:DayMins | 0.010466 | 1.010520 | 0.292434935 |
| AccountWeeks | 0.002182 | 1.002184 | 0.290022576 |
| AccountWeeks:DataPlan1 | 0.001355 | 1.001356 | 0.28978296 |
| DayMins:OverageFee | 0.000990 | 1.000990 | 0.289677043 |
| DayCalls:OverageFee | 0.000939 | 1.000940 | 0.289662574 |
| DayMins:MonthlyCharge | 0.000320 | 1.000320 | 0.289483152 |
| DataUsage:DayMins | -0.004010 | 0.995998 | 0.288232463 |
| CustServCalls:DayMins | -0.005153 | 0.994860 | 0.287903079 |
| DayMins | -0.017647 | 0.982508 | 0.284328585 |
| DayCalls | -0.018951 | 0.981228 | 0.283958049 |
| MonthlyCharge | -0.046615 | 0.954455 | 0.276210167 |
| DataPlan1:MonthlyCharge | -0.047739 | 0.953382 | 0.275899854 |
| CustServCalls:OverageFee | -0.059478 | 0.942256 | 0.272680008 |
| OverageFee | -0.069614 | 0.932754 | 0.269930045 |
| DataPlan1:OverageFee | -0.223311 | 0.799866 | 0.231473746 |
| ContractRenewal1:RoamMins | -0.251821 | 0.777384 | 0.224967552 |
| ContractRenewal1:DataPlan1 | -1.475645 | 0.228631 | 0.066163737 |
| ContractRenewal1 | -4.834198 | 0.007953 | 0.002301541 |
| Intercept | 0.9741334408 | 2.648871 |  |
|  |  | 345.553789 |  |

We can see that Customer Dataplan, CustomerService Call,Roammins, Datausage has highest importance in the model.

**Interpretation:**

**Dataplan:** This is factor variable.

*The ODDS of the customer to churn(1) when the customer has dataplan is 315.961 times than that of a customer without a dataplan.*

**Customer Service Calls:** This is a continuous variable.

*If Customer-service calls increase by one unit, keeping all the other independent variables constant, the likelihood that the customer will churn become 4.96 times higher than that of a non-churning customer.*

**CONCORDANCE TEST.**

Concordance is the percentage of predicted probability scores where the scores of actual positives are greater than the scores of actual negatives. To calculate we need to takes the scores of all possible pairs of ONES and ZEROS. 90% of concordant pairs based on the test sample.

> Concordance(actuals = testcell$Churn,predictedScores =predict.test )

$Concordance

[1] 0.9038478

$Discordance

[1] 0.09615218

$Tied

[1] 2.775558e-17

$Pairs

[1] 126799

**KS PLOT**

We can also do the KS plotting.

From the below plot we can see that by targeting 30% of the population, model is capable to capture 87.25% of total responder (Churn=1), while without the model we could have captured only 30% of responders by random targeting.

