

Logistic Regression Case Study3

Abhirup Sen

02/06/2021

```
#load data
```

```
churn_data <- read.csv("Churn Data.csv")
str(churn_data)
```

```
## 'data.frame':  3333 obs. of  10 variables:
## $ Churn          : int  0 0 0 0 0 0 0 0 0 0 ...
## $ AccountWeeks   : int  128 107 137 84 75 118 121 147 117 141 ...
## $ ContractRenewal: int  1 1 1 0 0 0 1 0 1 0 ...
## $ DataPlan       : int  1 1 0 0 0 0 1 0 0 1 ...
## $ DataUsage      : num  2.7 3.7 0 0 0 0 2.03 0 0.19 3.02 ...
## $ CustServCalls  : int  1 1 0 2 3 0 3 0 1 0 ...
## $ DayMins        : num  265 162 243 299 167 ...
## $ DayCalls       : int  110 123 114 71 113 98 88 79 97 84 ...
## $ MonthlyCharge  : num  89 82 52 57 41 57 87.3 36 63.9 93.2 ...
## $ RoamMins       : num  10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 8.7 11.2 ...
```

```
churn_data$Churn <- factor(churn_data$Churn)
str(churn_data)
```

```
## 'data.frame':  3333 obs. of  10 variables:
## $ Churn          : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
## $ AccountWeeks   : int  128 107 137 84 75 118 121 147 117 141 ...
## $ ContractRenewal: int  1 1 1 0 0 0 1 0 1 0 ...
## $ DataPlan       : int  1 1 0 0 0 0 1 0 0 1 ...
## $ DataUsage      : num  2.7 3.7 0 0 0 0 2.03 0 0.19 3.02 ...
## $ CustServCalls  : int  1 1 0 2 3 0 3 0 1 0 ...
## $ DayMins        : num  265 162 243 299 167 ...
## $ DayCalls       : int  110 123 114 71 113 98 88 79 97 84 ...
## $ MonthlyCharge  : num  89 82 52 57 41 57 87.3 36 63.9 93.2 ...
## $ RoamMins       : num  10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 8.7 11.2 ...
```

```
logit_model <- glm(Churn ~., data = churn_data, family = "binomial")
```

```
churn_data$churn_predicted <- predict(logit_model,data = churn_data, type ="response" )
```

```
churn_data$churn_predicted <- ifelse(churn_data$churn_predicted > 0.5,1,0)
```

```
table(churn_data$Churn, churn_data$churn_predicted)
```

```
##
##      0      1
## 0 2780    70
## 1   393    90
```

```
misClassError <- mean( churn_data$churn_predicted != churn_data$Churn)
print(paste('Accuracy =', 1 - misClassError))
```

```
## [1] "Accuracy = 0.861086108610861"
```

```
churn_data <- read.csv("Churn Data.csv")
str(churn_data)
```

```
## 'data.frame': 3333 obs. of 10 variables:
## $ Churn : int 0 0 0 0 0 0 0 0 0 0 ...
## $ AccountWeeks : int 128 107 137 84 75 118 121 147 117 141 ...
## $ ContractRenewal: int 1 1 1 0 0 0 1 0 1 0 ...
## $ DataPlan : int 1 1 0 0 0 0 1 0 0 1 ...
## $ DataUsage : num 2.7 3.7 0 0 0 0 2.03 0 0.19 3.02 ...
## $ CustServCalls : int 1 1 0 2 3 0 3 0 1 0 ...
## $ DayMins : num 265 162 243 299 167 ...
## $ DayCalls : int 110 123 114 71 113 98 88 79 97 84 ...
## $ MonthlyCharge : num 89 82 52 57 41 57 87.3 36 63.9 93.2 ...
## $ RoamMins : num 10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 8.7 11.2 ...
```

```
churn_data$Churn <- factor(churn_data$Churn)
str(churn_data)
```

```
## 'data.frame': 3333 obs. of 10 variables:
## $ Churn : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
## $ AccountWeeks : int 128 107 137 84 75 118 121 147 117 141 ...
## $ ContractRenewal: int 1 1 1 0 0 0 1 0 1 0 ...
## $ DataPlan : int 1 1 0 0 0 0 1 0 0 1 ...
## $ DataUsage : num 2.7 3.7 0 0 0 0 2.03 0 0.19 3.02 ...
## $ CustServCalls : int 1 1 0 2 3 0 3 0 1 0 ...
## $ DayMins : num 265 162 243 299 167 ...
## $ DayCalls : int 110 123 114 71 113 98 88 79 97 84 ...
## $ MonthlyCharge : num 89 82 52 57 41 57 87.3 36 63.9 93.2 ...
## $ RoamMins : num 10 13.7 12.2 6.6 10.1 6.3 7.5 7.1 8.7 11.2 ...
```

```
#split
```

```
library(caTools)
set.seed(100)
split <- sample.split(churn_data, SplitRatio = 0.8)
split
```

```
## [1] TRUE TRUE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE
```

```
train <- subset(churn_data,split ==TRUE)
test <- subset(churn_data,split == FALSE)

str(train)
```

```
## 'data.frame': 2666 obs. of 10 variables:
## $ Churn : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
## $ AccountWeeks : int 128 107 84 75 118 147 117 141 65 74 ...
## $ ContractRenewal: int 1 1 0 0 0 0 1 0 1 1 ...
## $ DataPlan : int 1 1 0 0 0 0 0 1 0 0 ...
## $ DataUsage : num 2.7 3.7 0 0 0 0 0.19 3.02 0.29 0.34 ...
## $ CustServCalls : int 1 1 2 3 0 0 1 0 4 0 ...
## $ DayMins : num 265 162 299 167 223 ...
## $ DayCalls : int 110 123 71 113 98 79 97 84 137 127 ...
## $ MonthlyCharge : num 89 82 57 41 57 36 63.9 93.2 44.9 49.4 ...
## $ RoamMins : num 10 13.7 6.6 10.1 6.3 7.1 8.7 11.2 12.7 9.1 ...
```

```
str(test)
```

```
## 'data.frame': 667 obs. of 10 variables:
## $ Churn : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
## $ AccountWeeks : int 137 121 168 85 130 57 172 36 34 65 ...
## $ ContractRenewal: int 1 1 1 1 1 1 1 1 1 1 ...
## $ DataPlan : int 0 1 0 1 0 1 0 1 0 0 ...
## $ DataUsage : num 0 2.03 0 3.73 0 2.57 0 3.92 0 0 ...
## $ CustServCalls : int 0 3 1 1 0 0 3 0 2 3 ...
## $ DayMins : num 243 218 129 196 183 ...
## $ DayCalls : int 114 88 96 139 112 115 121 128 82 120 ...
## $ MonthlyCharge : num 52 87.3 31 95.3 38 78.7 39 78.2 46 50 ...
## $ RoamMins : num 12.2 7.5 11.2 13.8 9.5 9.5 12.6 14.5 10 13.2 ...
```

```
#train
```

```
logit_model <- glm(Churn ~., data = train, family = "binomial")
summary(logit_model)
```

```
##
## Call:
## glm(formula = Churn ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.0160  -0.5140  -0.3527  -0.2169   2.9278
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -6.0090434  0.6081721  -9.880 < 2e-16 ***
## AccountWeeks    0.0008305  0.0015471   0.537 0.591399
## ContractRenewal -1.9339457  0.1594232 -12.131 < 2e-16 ***
## DataPlan       -0.7436052  0.5812937  -1.279 0.200818
## DataUsage      -0.8031518  0.2477746  -3.241 0.001189 **
## CustServCalls   0.5434821  0.0441639  12.306 < 2e-16 ***
```

```
## DayMins      -0.0009164  0.0027467  -0.334  0.738654
## DayCalls     0.0035212  0.0030571   1.152  0.249391
## MonthlyCharge 0.0763849  0.0148854   5.132  2.87e-07 ***
## RoamMins     0.0913726  0.0248226   3.681  0.000232 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 2212.2 on 2665 degrees of freedom
## Residual deviance: 1770.3 on 2656 degrees of freedom
## AIC: 1790.3
##
## Number of Fisher Scoring iterations: 5
```

```
logit_model <- glm(Churn ~ ContractRenewal + CustServCalls + RoamMins , data = train, family = "binomial")
summary(logit_model)
```

```
##
## Call:
## glm(formula = Churn ~ ContractRenewal + CustServCalls + RoamMins,
##      family = "binomial", data = train)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.6114  -0.5362  -0.4180  -0.3314   2.6104
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.88004    0.27788  -6.766 1.33e-11 ***
## ContractRenewal -1.83877    0.15009 -12.251 < 2e-16 ***
## CustServCalls   0.47305    0.04115  11.495 < 2e-16 ***
## RoamMins        0.07677    0.02164   3.548 0.000389 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 2212.2 on 2665 degrees of freedom
## Residual deviance: 1938.7 on 2662 degrees of freedom
## AIC: 1946.7
##
## Number of Fisher Scoring iterations: 5
```

#step 3: Predict

```
fitted.results <- predict(logit_model,test,type ="response")
fitted.results
```

```
##          3          7          13          17          23          27          33
## 0.05829330 0.15137174 0.08425200 0.10098516 0.04790310 0.04790310 0.20877026
##          37          43          47          53          57          63          67
```

##	0.06877664	0.11867843	0.21648113	0.13038867	0.06713256	0.16085752	0.23756543
##	73	77	83	87	93	97	103
##	0.15636889	0.05622152	0.18920465	0.22926798	0.06665338	0.06175765	0.05152852
##	107	113	117	123	127	133	137
##	0.11088082	0.28952526	0.10380742	0.17547500	0.24453004	0.07471379	0.27955282
##	143	147	153	157	163	167	173
##	0.14459704	0.10029031	0.06476824	0.07545023	0.18338625	0.06476824	0.18338625
##	177	183	187	193	197	203	207
##	0.05829330	0.11787780	0.07851483	0.10937614	0.04860825	0.14084009	0.15174822
##	213	217	223	227	233	237	243
##	0.08133732	0.05305016	0.05176112	0.15335467	0.22041310	0.11088082	0.06908143
##	247	253	257	263	267	273	277
##	0.05382680	0.16783490	0.07632165	0.08604603	0.23473929	0.15636889	0.26349396
##	283	287	293	297	303	307	313
##	0.09036729	0.10937614	0.05771386	0.11948375	0.05421913	0.06957677	0.05305016
##	317	323	327	333	337	343	347
##	0.10141714	0.04968448	0.08544421	0.62928038	0.05190501	0.06635853	0.09036729
##	353	357	363	367	373	377	383
##	0.05525930	0.36830746	0.08911308	0.24453004	0.28331327	0.12865757	0.19116674
##	387	393	397	403	407	413	417
##	0.16570138	0.40092522	0.04228992	0.07686461	0.03928758	0.03542585	0.26441665
##	423	427	433	437	443	447	453
##	0.05957053	0.09227842	0.09422577	0.06713256	0.06158822	0.17326458	0.04387219
##	457	463	467	473	477	483	487
##	0.11317180	0.07796119	0.18338625	0.05940672	0.08726121	0.04323261	0.28958973
##	493	497	503	507	513	517	523
##	0.44712343	0.10353536	0.48412817	0.13038867	0.11163993	0.10071966	0.57415401
##	527	533	537	543	547	553	557
##	0.05957053	0.12441540	0.10238767	0.96974980	0.42823146	0.17996233	0.05745613
##	563	567	573	577	583	587	593
##	0.07261842	0.12952066	0.35942288	0.22306273	0.08133732	0.31818646	0.11550397
##	597	603	607	613	617	623	627
##	0.17326458	0.05251994	0.06220398	0.22522368	0.43576634	0.12358150	0.17289040
##	633	637	643	647	653	657	663
##	0.12662197	0.07313713	0.05343715	0.07333584	0.16890986	0.04720766	0.05871616
##	667	673	677	683	687	693	697
##	0.11550397	0.05663026	0.05190501	0.19396048	0.17883217	0.19956410	0.06476824
##	703	707	713	717	723	727	733
##	0.10309538	0.04882835	0.05881691	0.48531437	0.09891336	0.06617737	0.09797217
##	737	743	747	753	757	763	767
##	0.37540688	0.38263430	0.05983715	0.21004123	0.11708187	0.21295778	0.26200682
##	773	777	783	787	793	797	803
##	0.04686356	0.05152852	0.07418479	0.05461415	0.31486483	0.06908143	0.03842746
##	807	813	817	823	827	833	837
##	0.05041447	0.09620987	0.08973821	0.08726121	0.33506866	0.11394459	0.04860825
##	843	847	853	857	863	867	873
##	0.48531437	0.07907206	0.17326458	0.09357260	0.21518181	0.05382680	0.08249200
##	877	883	887	893	897	903	907
##	0.06476824	0.06338707	0.04584549	0.09100034	0.15879577	0.55528542	0.06401959
##	913	917	923	927	933	937	943
##	0.12358150	0.14687302	0.07365925	0.06635853	0.07741112	0.04686356	0.03814479
##	947	953	957	963	967	973	977
##	0.14880736	0.18569680	0.09729581	0.15335467	0.17258534	0.08787460	0.06175765
##	983	987	993	997	1003	1007	1013

##	0.09554439	0.08787460	0.06570452	0.05581555	0.09529168	0.10569261	0.10353536
##	1017	1023	1027	1033	1037	1043	1047
##	0.07007540	0.04046317	0.11867843	0.07471379	0.20126350	0.24600917	0.03814479
##	1053	1057	1063	1067	1073	1077	1083
##	0.30664362	0.03987123	0.07686461	0.05663026	0.10452380	0.06027049	0.19880655
##	1087	1093	1097	1103	1107	1113	1117
##	0.14554919	0.07471379	0.10380742	0.26492546	0.17883217	0.06310570	0.19758656
##	1123	1127	1133	1137	1143	1147	1153
##	0.07963291	0.09865271	0.08787460	0.42519165	0.38375681	0.05027449	0.07108257
##	1157	1163	1167	1173	1177	1183	1187
##	0.08787460	0.05305016	0.20374309	0.11240359	0.25313913	0.23251444	0.04517851
##	1193	1197	1203	1207	1213	1217	1223
##	0.63758593	0.06203341	0.06000207	0.04860825	0.09292348	0.13625904	0.12747340
##	1227	1233	1237	1243	1247	1253	1257
##	0.13038867	0.07281584	0.05004821	0.55915097	0.34014892	0.10937614	0.09012691
##	1263	1267	1273	1277	1283	1287	1293
##	0.28014287	0.04618248	0.30719976	0.07524625	0.26343314	0.06476824	0.04720766
##	1297	1303	1307	1313	1317	1323	1327
##	0.10937614	0.07524625	0.18453873	0.05983715	0.10863054	0.44143722	0.09203347
##	1333	1337	1343	1347	1353	1357	1363
##	0.18338625	0.19038516	0.07885912	0.35765725	0.06220398	0.08484621	0.17658851
##	1367	1373	1377	1383	1387	1393	1397
##	0.20499138	0.06810054	0.18180345	0.11163993	0.04016616	0.12409681	0.70992382
##	1403	1407	1413	1417	1423	1427	1433
##	0.05688451	0.07159116	0.07418479	0.15236056	0.05647405	0.11708187	0.05421913
##	1437	1443	1447	1453	1457	1463	1467
##	0.08191281	0.06858936	0.12110857	0.13842502	0.07686461	0.10425008	0.03464722
##	1473	1477	1483	1487	1493	1497	1503
##	0.38089633	0.03676115	0.17107623	0.04860825	0.07578220	0.19758656	0.62680304
##	1507	1513	1517	1523	1527	1533	1537
##	0.12441540	0.05101268	0.06908143	0.05647405	0.14880736	0.05541235	0.04584549
##	1543	1547	1553	1557	1563	1567	1573
##	0.27558451	0.10452380	0.18039481	0.04825445	0.06131430	0.11472197	0.07796119
##	1577	1583	1587	1593	1597	1603	1607
##	0.18223937	0.07210310	0.06338707	0.13535803	0.04355128	0.55156821	0.05004821
##	1613	1617	1623	1627	1633	1637	1643
##	0.09662362	0.05305016	0.07524625	0.07007540	0.05421913	0.07159116	0.06476824
##	1647	1653	1657	1663	1667	1673	1677
##	0.04932325	0.06338707	0.08133732	0.82302193	0.09267700	0.15174822	0.10141714
##	1683	1687	1693	1697	1703	1707	1713
##	0.06957677	0.13842502	0.05940672	0.09163737	0.22522368	0.08726121	0.26194623
##	1717	1723	1727	1733	1737	1743	1747
##	0.07261842	0.09100034	0.08307492	0.15943120	0.06203341	0.08544421	0.09891336
##	1753	1757	1763	1767	1773	1777	1783
##	0.31054427	0.33165652	0.04106353	0.13213954	0.09227842	0.05343715	0.06713256
##	1787	1793	1797	1803	1807	1813	1817
##	0.04355128	0.03730872	0.09554439	0.42699233	0.05622152	0.31652330	0.20421848
##	1823	1827	1833	1837	1843	1847	1853
##	0.10937614	0.15335467	0.11041348	0.04860825	0.04584549	0.08604603	0.10596967
##	1857	1863	1867	1873	1877	1883	1887
##	0.07578220	0.21996420	0.11012623	0.42260186	0.06761494	0.51409528	0.13268512
##	1893	1897	1903	1907	1913	1917	1923
##	0.24600917	0.07686461	0.08425200	0.08249200	0.70349413	0.08787460	0.25313913
##	1927	1933	1937	1943	1947	1953	1957

##	0.11708187	0.07007540	0.13038867	0.10937614	0.05446318	0.05855455	0.20877026
##	1963	1967	1973	1977	1983	1987	1993
##	0.18223937	0.04342949	0.14026639	0.07963291	0.07491645	0.05787329	0.10071966
##	1997	2003	2007	2013	2017	2023	2027
##	0.06523482	0.21260018	0.14270826	0.05855455	0.16890986	0.08076552	0.10309538
##	2033	2037	2043	2047	2053	2057	2063
##	0.14687302	0.15636889	0.08425200	0.08366158	0.11629059	0.21778612	0.07907206
##	2067	2073	2077	2083	2087	2093	2097
##	0.07178600	0.14177160	0.68730193	0.04896451	0.09292348	0.12525412	0.07707262
##	2103	2107	2113	2117	2123	2127	2133
##	0.14212821	0.30238608	0.29110697	0.10380742	0.07872692	0.07578220	0.13716509
##	2137	2143	2147	2153	2157	2163	2167
##	0.08484621	0.09227842	0.26052511	0.36474251	0.19276308	0.06203341	0.12865757
##	2173	2177	2183	2187	2193	2197	2203
##	0.05421913	0.10211891	0.06175765	0.18802975	0.08581607	0.05541235	0.35238364
##	2207	2213	2217	2223	2227	2233	2237
##	0.08076552	0.05328925	0.13391029	0.05787329	0.08019739	0.05914190	0.15879577
##	2243	2247	2253	2257	2263	2267	2273
##	0.13038867	0.08484621	0.08076552	0.04538382	0.10669920	0.07741112	0.13302242
##	2277	2283	2287	2293	2297	2303	2307
##	0.13625904	0.20877026	0.05115460	0.12275241	0.07851483	0.13126163	0.05813278
##	2313	2317	2323	2327	2333	2337	2343
##	0.04686356	0.39731813	0.36823455	0.06635853	0.08019739	0.08665169	0.12029379
##	2347	2353	2357	2363	2367	2373	2377
##	0.08169290	0.06087394	0.08887571	0.09115112	0.10238767	0.11948375	0.05729778
##	2383	2387	2393	2397	2403	2407	2413
##	0.05305016	0.40954642	0.08307492	0.46999218	0.45537467	0.11240359	0.06476824
##	2417	2423	2427	2433	2437	2443	2447
##	0.13480316	0.16717264	0.09620987	0.30883606	0.09755326	0.12577540	0.13213954
##	2453	2457	2463	2467	2473	2477	2483
##	0.10029031	0.26343314	0.12609768	0.05152852	0.07851483	0.07686461	0.12029379
##	2487	2493	2497	2503	2507	2513	2517
##	0.03542585	0.04452078	0.06617737	0.18039481	0.09687949	0.05041447	0.13751197
##	2523	2527	2533	2537	2543	2547	2553
##	0.10642046	0.19961417	0.09797217	0.04618248	0.07471379	0.13751197	0.07159116
##	2557	2563	2567	2573	2577	2583	2587
##	0.12525412	0.06713256	0.15879577	0.07963291	0.12779939	0.10002644	0.08544421
##	2593	2597	2603	2607	2613	2617	2623
##	0.40832275	0.05729778	0.48531437	0.11818259	0.10642046	0.07652830	0.28169365
##	2627	2633	2637	2643	2647	2653	2657
##	0.14119473	0.08787460	0.09012691	0.34540793	0.38444944	0.04860825	0.11012623
##	2663	2667	2673	2677	2683	2687	2693
##	0.52941906	0.38270834	0.23617771	0.05787329	0.29117165	0.11867843	0.37368187
##	2697	2703	2707	2713	2717	2723	2727
##	0.20499138	0.06665338	0.06265332	0.05581555	0.13660399	0.11948375	0.11787780
##	2733	2737	2743	2747	2753	2757	2763
##	0.85602034	0.16890986	0.12441540	0.21909678	0.15738430	0.14026639	0.17216765
##	2767	2773	2777	2783	2787	2793	2797
##	0.35413759	0.20202805	0.04106353	0.19637222	0.45274585	0.08366158	0.07762048
##	2803	2807	2813	2817	2823	2827	2833
##	0.15879577	0.15076267	0.12694610	0.13694967	0.26498650	0.12029379	0.15137174
##	2837	2843	2847	2853	2857	2863	2867
##	0.12110857	0.04755417	0.07796119	0.05251994	0.04968448	0.05787329	0.12779939
##	2873	2877	2883	2887	2893	2897	2903

```
## 0.11240359 0.18223937 0.48531437 0.07686461 0.08787460 0.07261842 0.11867843
##      2907      2913      2917      2923      2927      2933      2937
## 0.03747967 0.03265073 0.12275241 0.12029379 0.19111828 0.08787460 0.04790310
##      2943      2947      2953      2957      2963      2967      2973
## 0.34540793 0.05064231 0.32650245 0.07007540 0.05566146 0.06713256 0.11948375
##      2977      2983      2987      2993      2997      3003      3007
## 0.37188686 0.11012623 0.12275241 0.08665169 0.09755326 0.16570138 0.07686461
##      3013      3017      3023      3027      3033      3037      3043
## 0.08366158 0.10669920 0.02942138 0.48715388 0.06617737 0.17107623 0.04387219
##      3047      3053      3057      3063      3067      3073      3077
## 0.10817152 0.14401060 0.12441540 0.06175765 0.14843689 0.04452078 0.11867843
##      3083      3087      3093      3097      3103      3107      3113
## 0.20250046 0.59843169 0.05004821 0.17547500 0.20877026 0.08665169 0.57602997
##      3117      3123      3127      3133      3137      3143      3147
## 0.07108257 0.22842985 0.05190501 0.10211891 0.07907206 0.14306683 0.10168427
##      3153      3157      3163      3167      3173      3177      3183
## 0.07471379 0.18223937 0.06310570 0.12029379 0.20499138 0.04323261 0.15435409
##      3187      3193      3197      3203      3207      3213      3217
## 0.17107623 0.08402639 0.09036729 0.14554919 0.07851483 0.03122678 0.02834484
##      3223      3227      3233      3237      3243      3247      3253
## 0.60762023 0.10002644 0.07686461 0.04136690 0.04419535 0.27558451 0.05622152
##      3257      3263      3267      3273      3277      3283      3287
## 0.27001145 0.07796119 0.17436703 0.14401060 0.07108257 0.15335467 0.06220398
##      3293      3297      3303      3307      3313      3317      3323
## 0.08726121 0.07210310 0.30827828 0.06761494 0.15675551 0.05871616 0.28014287
##      3327      3333
## 0.10891436 0.06494581
```

```
test$Churn
```

```
##      [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0
##     [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0
##     [75] 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 1 1
##    [112] 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
##    [149] 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
##    [186] 0 0 0 0 1 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
##    [223] 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0
##    [260] 0 1 1 0 0 1 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 1 0 0 0 0 0 1 0
##    [297] 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 1 0 0 1
##    [334] 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
##    [371] 0 0 1 1 0 0 0 0 1 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
##    [408] 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
##    [445] 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 1 0 0 1
##    [482] 0 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 1 0 0
##    [519] 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1
##    [556] 0 0 1 0 0 0 0 0 0 1 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 1 1 0 1 0
##    [593] 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0
##    [630] 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0
##    [667] 0
## Levels: 0 1
```

```
fitted.results.new <- fitted.results
```

```
#step 4 : change to probabilities
```



```
fitted.results.new <- ifelse(fitted.results.new >0.3,1,0)
fitted.results.new
```

```
##      3      7     13     17     23     27     33     37     43     47     53     57     63     67     73     77
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
##     83     87     93     97    103    107    113    117    123    127    133    137    143    147    153    157
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
##    163    167    173    177    183    187    193    197    203    207    213    217    223    227    233    237
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
##    243    247    253    257    263    267    273    277    283    287    293    297    303    307    313    317
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
##    323    327    333    337    343    347    353    357    363    367    373    377    383    387    393    397
##      0      0      1      0      0      0      0      1      0      0      0      0      0      0      1      0
##    403    407    413    417    423    427    433    437    443    447    453    457    463    467    473    477
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
##    483    487    493    497    503    507    513    517    523    527    533    537    543    547    553    557
##      0      0      1      0      1      0      0      0      1      0      0      0      1      1      0      0
##    563    567    573    577    583    587    593    597    603    607    613    617    623    627    633    637
##      0      0      1      0      0      1      0      0      0      0      0      1      0      0      0      0
##    643    647    653    657    663    667    673    677    683    687    693    697    703    707    713    717
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      1
##    723    727    733    737    743    747    753    757    763    767    773    777    783    787    793    797
##      0      0      0      1      1      0      0      0      0      0      0      0      0      0      1      0
##    803    807    813    817    823    827    833    837    843    847    853    857    863    867    873    877
##      0      0      0      0      0      1      0      0      1      0      0      0      0      0      0      0
##    883    887    893    897    903    907    913    917    923    927    933    937    943    947    953    957
##      0      0      0      0      1      0      0      0      0      0      0      0      0      0      0      0
##    963    967    973    977    983    987    993    997   1003   1007   1013   1017   1023   1027   1033   1037
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
##   1043   1047   1053   1057   1063   1067   1073   1077   1083   1087   1093   1097   1103   1107   1113   1117
##      0      0      1      0      0      0      0      0      0      0      0      0      0      0      0      0
##   1123   1127   1133   1137   1143   1147   1153   1157   1163   1167   1173   1177   1183   1187   1193   1197
##      0      0      0      1      1      0      0      0      0      0      0      0      0      0      1      0
##   1203   1207   1213   1217   1223   1227   1233   1237   1243   1247   1253   1257   1263   1267   1273   1277
##      0      0      0      0      0      0      0      0      1      1      0      0      0      0      1      0
##   1283   1287   1293   1297   1303   1307   1313   1317   1323   1327   1333   1337   1343   1347   1353   1357
##      0      0      0      0      0      0      0      0      1      0      0      0      0      1      0      0
##   1363   1367   1373   1377   1383   1387   1393   1397   1403   1407   1413   1417   1423   1427   1433   1437
##      0      0      0      0      0      0      0      1      0      0      0      0      0      0      0      0
##   1443   1447   1453   1457   1463   1467   1473   1477   1483   1487   1493   1497   1503   1507   1513   1517
##      0      0      0      0      0      0      1      0      0      0      0      0      1      0      0      0
##   1523   1527   1533   1537   1543   1547   1553   1557   1563   1567   1573   1577   1583   1587   1593   1597
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
##   1603   1607   1613   1617   1623   1627   1633   1637   1643   1647   1653   1657   1663   1667   1673   1677
##      1      0      0      0      0      0      0      0      0      0      0      0      1      0      0      0
##   1683   1687   1693   1697   1703   1707   1713   1717   1723   1727   1733   1737   1743   1747   1753   1757
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      1      1
##   1763   1767   1773   1777   1783   1787   1793   1797   1803   1807   1813   1817   1823   1827   1833   1837
##      0      0      0      0      0      0      0      0      1      0      1      0      0      0      0      0
##   1843   1847   1853   1857   1863   1867   1873   1877   1883   1887   1893   1897   1903   1907   1913   1917
##      0      0      0      0      0      0      1      0      1      0      0      0      0      0      1      0
##   1923   1927   1933   1937   1943   1947   1953   1957   1963   1967   1973   1977   1983   1987   1993   1997
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
```

```
## 2003 2007 2013 2017 2023 2027 2033 2037 2043 2047 2053 2057 2063 2067 2073 2077
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      1
## 2083 2087 2093 2097 2103 2107 2113 2117 2123 2127 2133 2137 2143 2147 2153 2157
##      0      0      0      0      0      1      0      0      0      0      0      0      0      0      1      0
## 2163 2167 2173 2177 2183 2187 2193 2197 2203 2207 2213 2217 2223 2227 2233 2237
##      0      0      0      0      0      0      0      0      1      0      0      0      0      0      0      0
## 2243 2247 2253 2257 2263 2267 2273 2277 2283 2287 2293 2297 2303 2307 2313 2317
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      1
## 2323 2327 2333 2337 2343 2347 2353 2357 2363 2367 2373 2377 2383 2387 2393 2397
##      1      0      0      0      0      0      0      0      0      0      0      0      0      1      0      1
## 2403 2407 2413 2417 2423 2427 2433 2437 2443 2447 2453 2457 2463 2467 2473 2477
##      1      0      0      0      0      0      1      0      0      0      0      0      0      0      0      0
## 2483 2487 2493 2497 2503 2507 2513 2517 2523 2527 2533 2537 2543 2547 2553 2557
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
## 2563 2567 2573 2577 2583 2587 2593 2597 2603 2607 2613 2617 2623 2627 2633 2637
##      0      0      0      0      0      0      1      0      1      0      0      0      0      0      0      0
## 2643 2647 2653 2657 2663 2667 2673 2677 2683 2687 2693 2697 2703 2707 2713 2717
##      1      1      0      0      1      1      0      0      0      0      1      0      0      0      0      0
## 2723 2727 2733 2737 2743 2747 2753 2757 2763 2767 2773 2777 2783 2787 2793 2797
##      0      0      1      0      0      0      0      0      0      1      0      0      0      1      0      0
## 2803 2807 2813 2817 2823 2827 2833 2837 2843 2847 2853 2857 2863 2867 2873 2877
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
## 2883 2887 2893 2897 2903 2907 2913 2917 2923 2927 2933 2937 2943 2947 2953 2957
##      1      0      0      0      0      0      0      0      0      0      0      0      1      0      1      0
## 2963 2967 2973 2977 2983 2987 2993 2997 3003 3007 3013 3017 3023 3027 3033 3037
##      0      0      0      1      0      0      0      0      0      0      0      0      0      1      0      0
## 3043 3047 3053 3057 3063 3067 3073 3077 3083 3087 3093 3097 3103 3107 3113 3117
##      0      0      0      0      0      0      0      0      0      1      0      0      0      0      1      0
## 3123 3127 3133 3137 3143 3147 3153 3157 3163 3167 3173 3177 3183 3187 3193 3197
##      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
## 3203 3207 3213 3217 3223 3227 3233 3237 3243 3247 3253 3257 3263 3267 3273 3277
##      0      0      0      0      1      0      0      0      0      0      0      0      0      0      0      0
## 3283 3287 3293 3297 3303 3307 3313 3317 3323 3327 3333
##      0      0      0      0      1      0      0      0      0      0      0
```

```
table(test$Churn, fitted.results.new)
```

```
##      fitted.results.new
##      0      1
## 0 534  38
## 1  65  30
```

```
misClassError <- mean(fitted.results.new != test$Churn)
print(paste('Accuracy =', 1 - misClassError))
```

```
## [1] "Accuracy = 0.845577211394303"
```

```
library(caret)
```

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

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

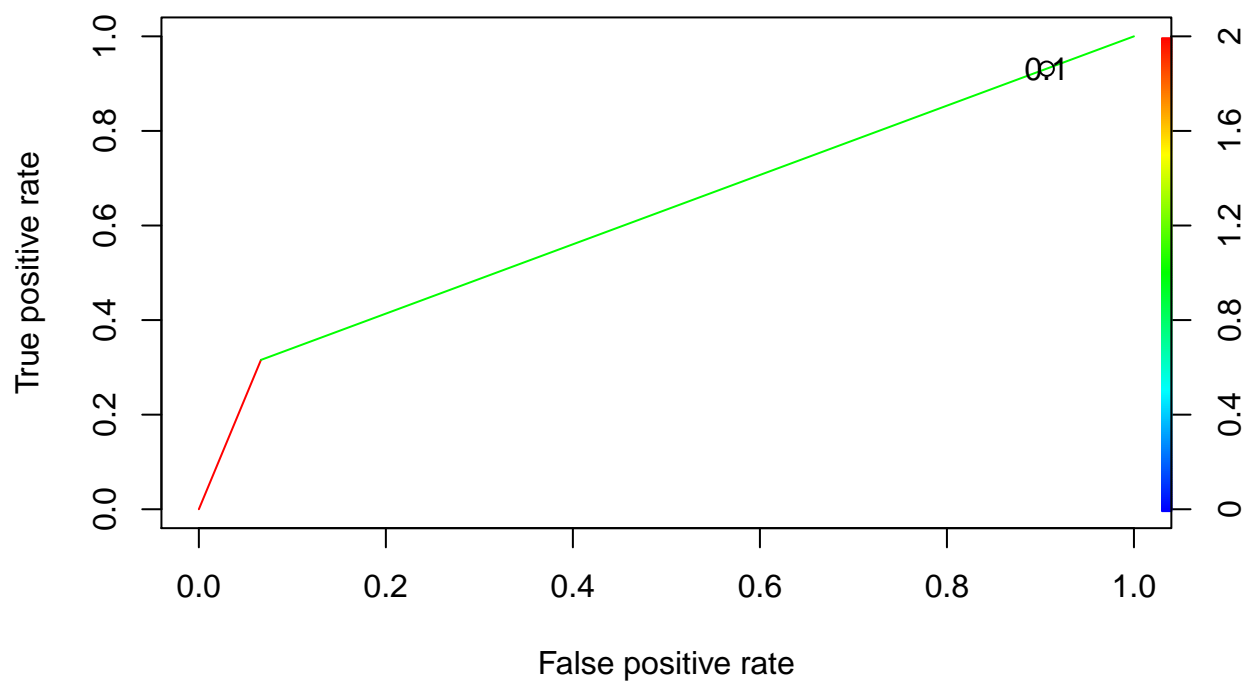
```
confusionMatrix(table(test$Churn, fitted.results.new))
```

```
## Confusion Matrix and Statistics
##
##      fitted.results.new
##      0    1
## 0 534  38
## 1  65  30
##
##              Accuracy : 0.8456
##              95% CI : (0.8159, 0.8722)
##      No Information Rate : 0.8981
##      P-Value [Acc > NIR] : 0.99999
##
##              Kappa : 0.2829
##
##  Mcnemar's Test P-Value : 0.01041
##
##      Sensitivity : 0.8915
##      Specificity : 0.4412
##      Pos Pred Value : 0.9336
##      Neg Pred Value : 0.3158
##      Prevalence : 0.8981
##      Detection Rate : 0.8006
##      Detection Prevalence : 0.8576
##      Balanced Accuracy : 0.6663
##
##      'Positive' Class : 0
##
```

```
library(ROCR)
```

```
ROCRPred <- prediction(fitted.results.new, test$Churn)
ROCRPerf <- performance(ROCRPred, measure = "tpr", x.measure = "fpr")

#plot(str(ROCRPred))
plot(ROCRPerf, colorize = TRUE)
plot(ROCRPerf, colorize = TRUE, print.cutoffs.at = seq(0.1, 0.1))
```



```
plot(ROCRPerf, colorize = TRUE, print.cutoffs.at = seq(0.1, 0.1), main = "ROC Curve")
abline(a=0, b=1)
```

```
auc <- performance(ROCRPred, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.624678
```

```
auc <- round(auc,4)
legend(.5,.4, auc, title = "AUC")
```

