Логистическая регрессия в R на примере данных оттока клиентов из телекоммуникационной компании

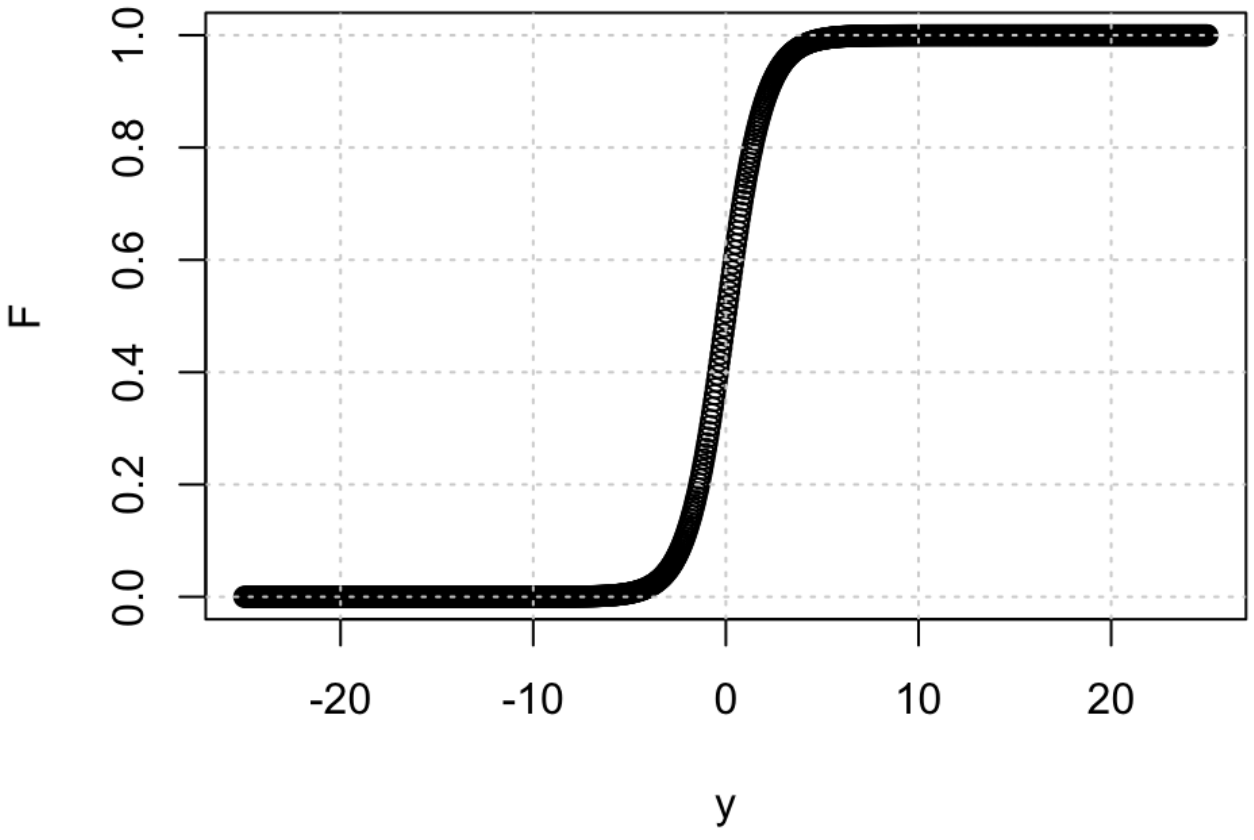
Постановка задачи

При решении задач бинарной классификации (вероятность имеет два значения 0 или 1) зачастую используют логистическую регрессию. Из определения — логистическая регрессия — это статистическая модель, которая применяется для предсказания вероятности возникновения некоторого события по значениям множества признаков.

$$ P(y)=\frac{1}{1+e^{-F(y)}} $$

где $□(y)$ — некоторая (в общем случая не линейная) функция от исследуемых переменных $y_□$ вида: $□(y)=□_0+□_1·y_1+⋯+□_□·y_□$.

Значение □(y) — есть вероятность события $(0 \leq □(y) \leq 0)$, тогда $□(y)$ — некоторая функция, принимающая значения от $-\infty$ до $+\infty$.



Как видно из графика, для большинства значений аргумента вероятность события (y) может принимать одно из двух значений 0 или 1, однако вблизи точки x=0 значение испытывает скачок. Здесь чтобы разделить исходы на 0 и 1 мы должны ввести точку отсечки — значение вероятности события (y) выше которой трактуются как 1, а ниже — как 0. По умолчанию, это значение равно 0.5. Тем не менее, в каждом конкретном случае мы модем изменять её для обеспечения лучших результатов в классификации. Например, если нас интересует только один из возможных исходов.

Описательный анализ

Ниже по ссылке лежат исходные данные для анализа. Их необходимо скачать и распаковать файл в директорию с проектом.

```
# нам понядобятся следующие пакеты
library(tidyverse)
library(knitr)
library(ROCR)
library(caret)

# обратите внимание на то, как мы читаем данные:
# указываем, чтобы первая строка в файле трактовалась как строка заголовка
# также говорим чтобы при чтении строковые значения воспринимались как категориальные
teleco.raw.data <- read.csv("WA_Fn-UseC_-Telco-Customer-Churn.csv", header = TRUE, stringsAsFactors = TRUE)
```

customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No	No	No	No	Month-to-month	Yes	Electronic check	29.85	29.85	No
5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	No	No	No	One year	No	Mailed check	56.95	1889.50	No
3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	No	No	No	Month-to-month	Yes	Mailed check	53.85	108.15	Yes
7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes	Yes	No	No	One year	No	Bank transfer (automatic)	42.30	1840.75	No
9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	No	No	No	Month-to-month	Yes	Electronic check	70.70	151.65	Yes

При загрузке файла система сама определила тип каждого из столбцов, но во избежание проблем мы должны проверить их

```
str(teleco.raw.data)

## 'data.frame':    7043 obs. of  21 variables:
##  $ customerID      : Factor w/ 7043 levels "0002-ORFBO","0003-MKNFE",..:
5376 3963 2565 5536 6512 6552 1003 4771 5605 4535 ...
##  $ gender          : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1
1 2 ...
##  $ SeniorCitizen   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1
...
##  $ Dependents      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 1 1 2
...
##  $ tenure          : int  1 34 2 45 2 8 22 10 28 62 ...
##  $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 1 2 2
...
##  $ MultipleLines   : Factor w/ 3 levels "No","No phone service",..: 2 1 1
```

```
2 1 3 3 2 3 1 ...
##  $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",..: 1 1 1 1 2
2 2 1 2 1 ...
##  $ OnlineSecurity  : Factor w/ 3 levels "No","No internet service",..: 1
3 3 3 1 1 1 3 1 3 ...
##  $ OnlineBackup    : Factor w/ 3 levels "No","No internet service",..: 3
1 3 1 1 1 3 1 1 3 ...
##  $ DeviceProtection: Factor w/ 3 levels "No","No internet service",..: 1
3 1 3 1 3 1 1 3 1 ...
##  $ TechSupport     : Factor w/ 3 levels "No","No internet service",..: 1
1 1 3 1 1 1 1 3 1 ...
##  $ StreamingTV     : Factor w/ 3 levels "No","No internet service",..: 1
1 1 1 1 3 3 1 3 1 ...
##  $ StreamingMovies : Factor w/ 3 levels "No","No internet service",..: 1
1 1 1 1 3 1 1 3 1 ...
##  $ Contract        : Factor w/ 3 levels "Month-to-month",..: 1 2 1 2 1 1
1 1 1 2 ...
##  $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 1 2 1
...
##  $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",..: 3
4 4 1 3 3 2 4 3 1 ...
##  $ MonthlyCharges  : num  29.9 57 53.9 42.3 70.7 ...
##  $ TotalCharges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
##  $ Churn           : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 2 1 1 2 1
...
```

Ага, фактор `SeniorCitizen` помечен как числовой, но очевидно, что он категориальный, заменим его. Также переместим фактор `Churn` на первое место для удобства.

```
teleco.raw.data$SeniorCitizen=as.factor(teleco.raw.data$SeniorCitizen)
teleco.raw.data %>%
  relocate(Churn) -> prepared.data
```

Теперь посмотрим, в каких столбцах находятся пропущенные данные и сколько их:

```
sapply(prepared.data,function(x) sum(is.na(x)))
##           Churn           gender    SeniorCitizen          Partner
##               0                0                0                0
##      Dependents           tenure     PhoneService    MultipleLines
##               0                0                0                0
## InternetService   OnlineSecurity     OnlineBackup DeviceProtection
##               0                0                0                0
##     TechSupport      StreamingTV  StreamingMovies         Contract
##               0                0                0                0
##PaperlessBilling    PaymentMethod   MonthlyCharges     TotalCharges
##               0                0                0               11
```

Замечательно, пустые значения есть только в столбце `TotalCharges` и их всего 11. Потому

можем просто удалить записи, которые содержат пустые значения.

```
teleco.raw.data[complete.cases(teleco.raw.data),]->teleco.raw.data
# мы воспользовались функцией complete.cases, котора возвращает
# логический индекс строк, которые не содержат пустые значения

prepared.data[complete.cases(prepared.data),] ->prepared.data
```

Подготовим тренировочную и тестовую выборки из наших данных

Соотношение числа элементов тренировочной и тестовой выборки будет 75%:25%.

```
#Предварительно закодируем наши факторы как серию столбцов из 0 и 1, так называемое
`one_hot` кодирование.
require(mltools)


one_hot(as.data.table(prepared.data), dropUnusedLevels=T)->prepared.data

smp_size <- floor(0.75 * nrow(prepared.data))
set.seed(123)
train_ind <- sample(seq_len(nrow(prepared.data)), size = smp_size)

train_df <- prepared.data[train_ind, ]
test_df <- prepared.data[-train_ind, ]
```

Теперь у нас есть тренировочный и тестовый наборы, определим модель:

```
model <- glm(Churn ~.,family=binomial(link='logit'),data=train_df)

# Вариационный анализ покажет нам наиболее важные факторы
anova(model, test="Chisq")->res
res[with(res, order(-Deviance)),]

##
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Churn
##
## Terms added sequentially (first to last)
##
##
##                                          Df Deviance Resid. Df Resid.
```

```
Dev
## tenure                                    1   637.02      5268
5217.6
## `InternetService_Fiber optic`             1   370.10      5264
4644.5
## MultipleLines_No                          1   143.90      5266
5073.2
## Partner_No                                1   124.20      5270
5891.6
## SeniorCitizen_0                           1    91.70      5271
6015.8
## `Contract_Month-to-month`                 1    80.86      5257
4479.0
## InternetService_DSL                       1    58.65      5265
5014.6
## Dependents_No                             1    36.94      5269
5854.6
## OnlineSecurity_No                         1    28.75      5263
4615.7
## StreamingTV_No                            1    23.88      5259
4571.4
## TechSupport_No                            1    19.90      5260
4595.3
## TotalCharges                              1    15.58      5250
4415.5
## `Contract_One year`                       1    14.79      5256
4464.2
## PaperlessBilling_No                       1    12.68      5255
4451.5
## StreamingMovies_No                        1    11.60      5258
4559.8
## `PaymentMethod_Electronic check`          1    10.46      5252
4433.3
## `PaymentMethod_Credit card (automatic)`   1     7.06      5253
4443.7
## MonthlyCharges                            1     2.20      5251
4431.1
## `PaymentMethod_Bank transfer (automatic)` 1     0.70      5254
4450.8
## gender_Female                             1     0.57      5272
6107.5
## PhoneService_No                           1     0.47      5267
5217.1
## OnlineBackup_No                           1     0.27      5262
4615.5
## DeviceProtection_No                       1     0.26      5261
4615.2
```

Анализ результатов

Теперь давайте посмотрим, как наша модель работает на тестовой выборке:

```
fitted.results <- predict(model,newdata=test_df,type='response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)
misClasificError <- mean(fitted.results != test_df$Churn)
print(paste('Accuracy',1-misClasificError))

## [1] "Accuracy 0.82"
```

Здесь мы получили, что модель дает правильные результаты в 82% случаев, при этом это процент верно предсказанных позитивных исходов (в нашем случае это сигнал, что клиент остался с поставщиком) Sensitivity=TP/(TP+FN) И его можно оценить из матрицы неточностей (см. ниже)

```
# матрица неточностей имеет вид
confusionMatrix(data=factor(fitted.results), reference =
factor(test_df$Churn))

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1180  196
##          1  111  271
##
##                Accuracy : 0.8254
##                  95% CI : (0.8068, 0.8428)
##     No Information Rate : 0.7344
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5248
##
##  Mcnemar's Test P-Value : 1.634e-06
##
##             Sensitivity : 0.9140
##             Specificity : 0.5803
##          Pos Pred Value : 0.8576
##          Neg Pred Value : 0.7094
##              Prevalence : 0.7344
##          Detection Rate : 0.6712
##    Detection Prevalence : 0.7827
##       Balanced Accuracy : 0.7472
##
##        'Positive' Class : 0
```

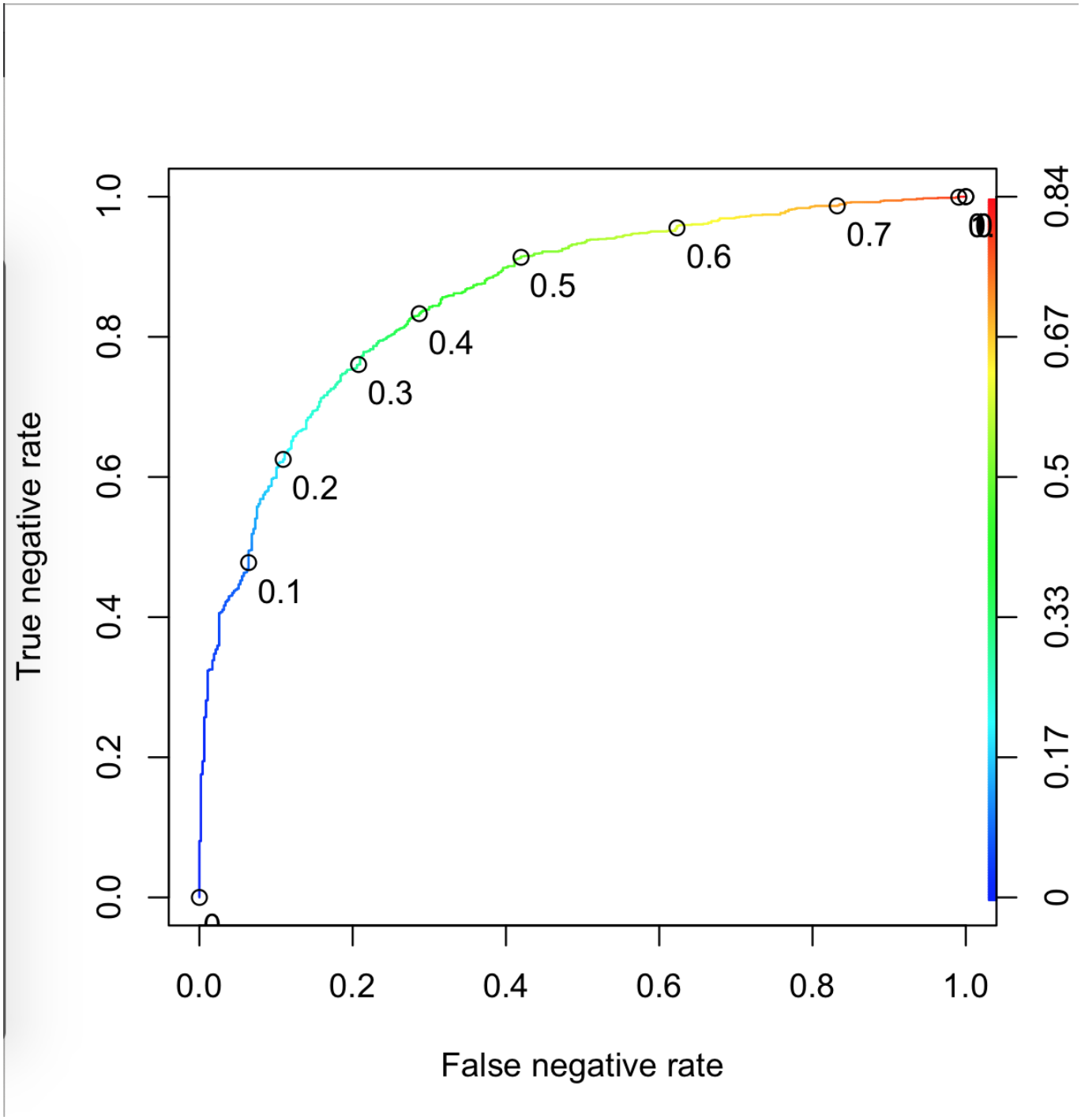
Здесь $Accuracy=\frac{TP+TN}{P+N}$, $Senitivity = \frac{TP}{P}$, $Specificity=\frac{TN}{N}$,

$Precisiob=\frac{TP}{TP+FP}$.

В нашем случае Specificity=58% — специфичность - процент верно предсказанных негативных исходов, то есть случаев, когда клиент ушел от поставщика услуг. В то же время Sensitivity=91%, число верно предсказанных положительных исходов, тоесть случаев, когда клиент остался с поставщиком услуг.

Общая точность модели составляет 82%.

Построим теперь график зависимости True negative rate от False negative rate.



Здесь на графике мы видим, что увеличение порога отсечки увеличивает как TPR, так и TNR. Здесь варе баланс. В зависимости от того, предсказание какого исход предпочтительнее, выбирается та или иная точка отсечки.

Вывод

Что в итоге: мы натренировали модель логистической регрессии, которая по имеющимся данным предсказывает уйдет ли клиент в другую компанию, или останется здесь. Полученная модель в 58% случаев верно указывает на такие ситуации. Модель обучена не идеально, выбор точки отсечки может существенно улучшить предсказательную способность модели к тому или иному исходу.