LOGISTIC REGRESSION

Chapter 8: Logistic Regression

- Odds and Probability
- Logistic Regression
- Logistic Regression in R
- Coefficients
- General Example

Dalgaard, Chapter 13

Odds

The *odds* of an event E is defined as

$$\frac{P(E)}{1 - P(E)}.$$

AFL football on Anzac Day: Collingwood vs Essendon

$P(\text{Collingwood wins}) = 4/5$

$P(\text{Collingwood does not win}) = 1/5$

$$\text{Odds of Collingwood winning} = \frac{4/5}{1/5} = 4$$

"Odds of 4 to 1"

— the chance of Collingwood winning is 4 times the chance of Essendon winning (i.e. Collingwood losing).

A linear model for binary response

- Often the response variable of interest in data is a binary one (takes the value 0 or 1).
- For example, we may wish to model the incidence of hypertension (1=hypertension, 0=none) as a response variable with smoking, age, gender as explanatory variables.
- Denote these variables by Y, x_1, x_2, x_3.

A linear model for binary response

- Now x_2 (age) is a "continuous variable", but incidence of hypertension takes only 2 values, so it is more natural to try to model the *probability of observing the outcome 1*.
- A probability lies between 0 and 1, so we will fit a non-linear model that lives between 0 and 1, and then transform it to get a linear model.

$$p_i := Pr[Y = 1|x'es] = g(\alpha + \beta_1 x_1 + ... + \beta_k x_k)$$

  $g()$ forces the predicted values to be between 0 and 1.
- A standard choice called **logistic regression**, uses the fact that e^x is non-negative for every x, and that therefore

$$g(x_1, ..., x_k) := \frac{e^{\alpha+\beta_1 x_1+...+\beta_k x_k}}{1 + e^{\alpha+\beta_1 x_1+...+\beta_k x_k}} \in (0, 1).$$

Logistic Regression

Notice that

$$\frac{g(x)}{1-g(x)} = e^x,$$

and therefore

$$\log\left(\frac{g(x)}{1-g(x)}\right) = x.$$

Logistic Regression

Writing p_i for the probability that the ith observation is a 1, our model is that

$$p_i \approx \frac{e^{\alpha+\sum_{j=1}^{k}\beta_j x_{i,j}}}{1+e^{\alpha+\sum_{j=1}^{k}\beta_j x_{i,j}}},$$

if we have k explanatory variables.

With this model we have

$$\log\left(\frac{p_i}{1-p_i}\right) \approx \alpha + \sum_{j=1}^{k}\beta_j x_{i,j},$$

a linear model!

Recall: $\frac{p_i}{1-p_i}$ is the odds !!!
So we have a linear relationship between log-odds and x'es.

Logistic Regression in R

To fit such a model in R, we use the function glm in the form

```
> glm(y~x_1+x_2,family=binomial())
```

Hypertension Example - Dalgaard

```
> summary(glm(hyp~smoking+obesity+snoring,binomial))

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.37766    0.38018  -6.254    4e-10 ***
smokingYes  -0.06777    0.27812  -0.244   0.8075
obesityYes   0.69531    0.28509   2.439   0.0147 *
snoringYes   0.87194    0.39757   2.193   0.0283 *
---
    Null deviance: 14.1259  on 7  degrees of freedom
Residual deviance:  1.6184  on 4  degrees of freedom
AIC: 34.537

Number of Fisher Scoring iterations: 4
```

Interpreting coefficients

- By Looking directly on the coefficients, we interpret the *sign* of it but not the *magnitude*.
- An increase in x makes the outcome of $y = 1$ *more or less likely*.

Interpreting coefficients

The **intercept** represents the value of $\log(p/(1-p))$ when the explanatory variables are set to 0 (for factors this means set to the baseline level).

E.g. in the above example the model says that for a person who does not smoke, is not obese, and who does not snore, $\log(p/(1-p)) \approx -2.38$.

This means that for these values of explanatory variables

$$\frac{p}{1-p} \approx e^{-2.38} \approx 0.09.$$

This means that the *odds* of a person with these values of explanatory variables having hypertension are about 0.09 to 1, or equivalently the probability of having hypertension with these values is about

$$\frac{.09}{1+.09} \approx 0.08.$$

Interpreting coefficients

Note,

- To convert from a probability to odds, divide the probability by one minus that probability.

$$odds = \frac{p}{1-p}$$

- To convert from odds to a probability, divide the odds by one plus the odds.

$$p = \frac{odds}{1+odds}$$

Interpreting coefficients

The remaining coefficients in this example indicate differences from the baseline.

E.g. a non-smoking, non-snoring obese person has odds of having hypertension of about

$$e^{-2.38+0.695},$$

or in other words, $e^{0.695}$ times the baseline odds.

In other words, taking the exponential of the coefficients gives you a multiplicative factor for the odds.

For this reason (interpretation of coefficients) people often talk about the odds coming from these models rather than the probabilities themselves.

For a continuous explanatory variable x_1, the coefficient β_1 tells you (roughly speaking) that if you increase x_1 by 1 unit, the odds of getting 1 for the outcome are multiplied by e^{β_1}.

Churn Modelling

Customer Churn refers to when a customer (player, subscriber, user, etc.) stop his or her relationship with a company. Telecommunications companies have a high interest in understanding and predicting their customer churn.

We will use a dataset from IBM Sample Data Sets, which is available for free online. There is also an online analysis of this data, which I used, and contains other methods of analysing the same dataset.

(https://datascienceplus.com/
predict-customer-churn-logistic-regression-decision-tree-and-random-forest/)

Data Exploration and Processing

```
> churn <- read.csv('Telco-Customer-Churn.csv')
> str(churn)
```

```
'data.frame': 7043 obs. of  21 variables:
 $ customerID      : Factor w/ 7043 levels "0002-ORFBO","0003-MKNFE",..: 5376 3963 2565 5536 6512 6552 10
 $ 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 ...
```

Data Exploration and Cleaning

Find missing values and decide what to do with them.

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

```
> churn <- churn[complete.cases(churn), ]
```

Further cleaning may involve:

- grouping variables together;
- changing names;
- changing levels in some factors;
- etc.

We will be using our domain knowledge to make those decisions.

```
> ls(churn)  #list of variables
> summary(churn)
```

Data Exploration and Processing

```
      customerID      gender     SeniorCitizen     Partner     Dependents       tenure
 0002-ORFBO:    1   Female:3483   Min.   :0.0000   No :3639   No :4933   Min.   : 1.00
 0003-MKNFE:    1   Male  :3549   1st Qu.:0.0000   Yes:3393   Yes:2099   1st Qu.: 9.00
 0004-TLHLJ:    1                 Median :0.0000                         Median :29.00
 0011-IGKFF:    1                 Mean   :0.1624                         Mean   :32.42
 0013-EXCHZ:    1                 3rd Qu.:0.0000                         3rd Qu.:55.00
 0013-MHZWF:    1                 Max.   :1.0000                         Max.   :72.00
 (Other)   :7026

PhoneService          MultipleLines     InternetService            OnlineSecurity
 No : 680     No              :3385   DSL        :2416   No                 :3497
 Yes:6352     No phone service: 680   Fiber optic:3096   No internet service:1520
              Yes             :2967   No         :1520   Yes                :2015

              OnlineBackup              DeviceProtection              TechSupport
 No                 :3087   No                 :3094   No                 :3472
 No internet service:1520   No internet service:1520   No internet service:1520
 Yes                :2425   Yes                :2418   Yes                :2040

              StreamingTV              StreamingMovies            Contract
 No                 :2809   No                 :2781   Month-to-month:3875
 No internet service:1520   No internet service:1520   One year      :1472
 Yes                :2703   Yes                :2731   Two year      :1685
```

Data Exploration and Processing

```
PaperlessBilling                    PaymentMethod  MonthlyCharges    TotalCharges
 No :2864        Bank transfer (automatic):1542   Min.   : 18.25   Min.   :  18.8
 Yes:4168        Credit card (automatic)  :1521   1st Qu.: 35.59   1st Qu.: 401.4
                 Electronic check         :2365   Median : 70.35   Median :1397.5
                 Mailed check             :1604   Mean   : 64.80   Mean   :2283.3
                                                  3rd Qu.: 89.86   3rd Qu.:3794.7
                                                  Max.   :118.75   Max.   :8684.8

Churn
 No :5163
 Yes:1869
```

For the continuous data we want to look at possible correlations.

```
> #Changes to variables
> churn$SeniorCitizen <- as.factor(churn$SeniorCitizen)
> #Correlations
> numeric.var <- sapply(churn, is.numeric)
> corr.matrix <- cor(churn[,numeric.var])
```

Data Exploration and Processing

We obtain the correlation matrix.

	tenure	MonthlyCharges	TotalCharges
tenure	1.0000000	0.2468618	0.8258805
MonthlyCharges	0.2468618	1.0000000	0.6510648
TotalCharges	0.8258805	0.6510648	1.0000000

We can see that tenure and TotalCharges are very highly correlated, so we drop TotalCharges from our dataset. It is also a bit redundant to have both MonthlyCharges and TotalCharges in our model.

Data Exploration and Processing

We have complete dependence between some factor levels e.g. StreamingTV="No internet service" and StreamingMovies="No internet service" are identical. A statistical algorithm will not be able to distinguish the effect of changing one of these or the other (since they are identical).

```
> phone0=which(churn$PhoneService=="No")
> churn[phone0,8]="No"  #used to be "No phone service"
> internet0=which(churn$InternetService=="No")
> churn[internet0,10:15]="No"
```

Data Exploration and Processing

We also changed the Churn variable to avoid confusion later.

```
> churn$Churn=as.character(churn$Churn)
> churn$Churn[churn$Churn=="No"]="0"
> churn$Churn[churn$Churn=="Yes"]="1"
> churn$Churn=as.numeric(churn$Churn)
```

Now, we create the final dataset in which we exclude customerID and TotalCharges.

```
> churn.final = churn[,-c(1,20)]
```

Logistic Regression Model

The first thing we do is split the data into a **training set** and a **testing set**. We do this to be able to test our model on the testing set.

- We might not always have enough data.
- Depending on the data, we might need to do this carefully, making sure that we don't miss a type of customer from the training set.
- If the data is a time series, we can try and predict the last time period. E.g. If we have 4 years of data, we can train on the first 3 and predict the fourth.

Logistic Regression Model

In our case, we can just sample directly.

```
> xx <- sample(1:nrow(churn.final), 5000, replace = FALSE)
> train=churn.final[xx,]
> test=churn.final[-xx,]
```

And then run our logistic regression on the train data.

```
> churn.lg = glm(Churn ~., train,family=binomial() )
> summary(churn.lg)
```

Logistic Regression Model

```
glm(formula = Churn ~ ., family = binomial(), data = train)
Coefficients:
                                     Estimate Std. Error z value Pr(>|z|)
(Intercept)                         -0.039915   0.957594  -0.042 0.966752
genderMale                          -0.049102   0.076880  -0.639 0.523030
SeniorCitizen1                       0.116004   0.100665   1.152 0.249163
PartnerYes                           0.011281   0.092600   0.122 0.903034
DependentsYes                       -0.193059   0.105997  -1.821 0.068552 .
tenure                              -0.031767   0.002796 -11.362  < 2e-16 ***
PhoneServiceYes                     -0.373525   0.769501  -0.485 0.627384
MultipleLinesYes                     0.416027   0.209293   1.988 0.046837 *
InternetServiceFiber optic           0.951159   0.944365   1.007 0.313841
InternetServiceNo                   -1.018571   0.956439  -1.065 0.286893
OnlineSecurityYes                   -0.341631   0.209856  -1.628 0.103539
OnlineBackupYes                     -0.142196   0.207626  -0.685 0.493429
DeviceProtectionYes                 -0.086797   0.209192  -0.415 0.678205
TechSupportYes                      -0.375597   0.215108  -1.746 0.080797 .
StreamingTVYes                       0.309537   0.388407   0.797 0.425487
StreamingMoviesYes                   0.333748   0.387086   0.862 0.388575
ContractOne year                    -0.643422   0.125946  -5.109 3.24e-07 ***
ContractTwo year                    -1.161219   0.197317  -5.885 3.98e-09 ***
PaperlessBillingYes                  0.344545   0.088649   3.887 0.000102 ***
PaymentMethodCredit card (automatic) -0.101140   0.133962  -0.755 0.450254
PaymentMethodElectronic check        0.277345   0.113253   2.449 0.014329 *
PaymentMethodMailed check           -0.009980   0.136095  -0.073 0.941544
MonthlyCharges                      -0.004922   0.037566  -0.131 0.895764

    Null deviance: 5747.2  on 4999  degrees of freedom
Residual deviance: 4175.5  on 4977  degrees of freedom
AIC: 4221.5

Number of Fisher Scoring iterations: 6
```

Logistic Regression Model - Testing

Now that we have a model of choice, we can use it for predictions. First, let us test its prediction power on our test data.

	$y = 1$	$y = 0$
$\hat{y} = 1$		
$\hat{y} = 0$		

```
> prediction.test = predict.glm(churn.good, newdata = test,
type = 'response')
> print("Accuracy Matrix for Logistic Regression");
table(test$Churn, prediction.test > 0.5)
```

```
[1] "Accuracy Matrix for Logistic Regression"

    FALSE TRUE
  0  1322  149
  1   263  298
```

Percentage of correctly predicted values $\frac{1322+298}{1322+149+298+298} = 0.78$

Logistic Regression Model - Testing

```
> prediction.test = predict.glm(churn.good, newdata = test,
type = 'response')
> print("Accuracy Matrix for Logistic Regression");
table(test$Churn, prediction.test > 0.75)

[1] "Accuracy Matrix for Logistic Regression"
   FALSE TRUE
  0  1461   10
  1   502   59
  Percentage of correctly predicted values $\frac{1461+59}{1461+10+502+59} = 0.748$
```

```
> prediction.test = predict.glm(churn.good, newdata = test,
type = 'response')
> print("Accuracy Matrix for Logistic Regression");
table(test$Churn, prediction.test > 0.72)

[1] "Accuracy Matrix for Logistic Regression"
   FALSE TRUE
  0  1449   22
  1   483   78
 Percentage of correctly predicted values $\frac{1449+78}{1449+22+483+78} = 0.751$
```

Considering the type of information we have about the customers, we can't expect a very strong prediction power. There are two classical ways in which this prediction power is presented in industry.

Logistic Regression Model - ROC

ROC curves (receiver operating characteristic curves)

- Commonly used to characterise the diagnostic ability of a binary classifier.
- Most statistical classifiers produce prediction in the range between 0 and 1.
- Turning these values into yes or no predictions requires setting a *threshold*.
- Cases with predictors above the threshold are classified as positive, and cases with predictors below the threshold are predicted to be negative.

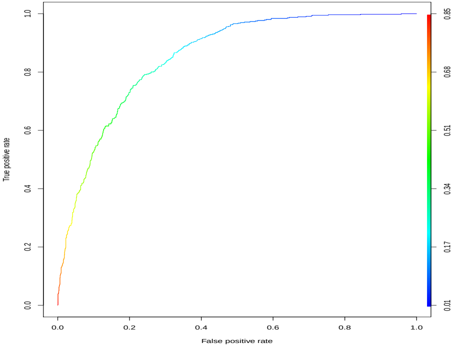
Logistic Regression Model - ROC

ROC curves (receiver operating characteristic curves)

- A **high threshold** is more conservative about labelling a case as positive; this makes *it less likely to produce false positive results* but *more likely to miss cases that are in fact positive* (lower rate of true positives).
- A **low threshold** produces positive labels more liberally, so it is less specific (*more false positives*) but also more sensitive (*more true positives*).
- The ROC curve is created by plotting the *true positive* rate (TPR) against the *false positive* rate (FPR).

Logistic Regression Model - ROC

In our case we have the following



and $AUC = 0.85$. (Area under the ROC Curve)
A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0