Chapter 4: Classification

Esben Eickhardt

2023-07-04

Introduction

In this chapter we focus on *classfication*. This includes *logistic regression*, *linear discriminant analysis*, *quadratic discriminant analysis*, *naive Bayes* and *K-nearest neighbors*.

All the same things can also be learned by following the following Statquest playlist.

Data

In this chapter we will try to predict if a person will default or not based on income and account balance (see figure).

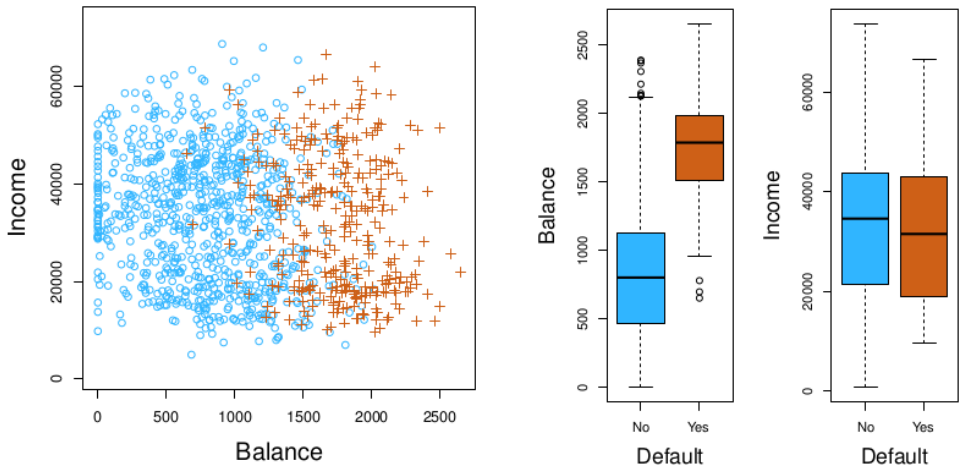


Figure 1: dataset

Why not Linear Regression?

Unfortunately the coding of output variable would imply an ordering on the outcomes, which would mess up the model. Even for two classes a model would not output probabilities, but it would also output values outside the interval [0:1].

1. Logistic Regression

Instead of fitting a line to the data, a *logistic regression* fits an S-shaped logistic function with the max and minimum values 1 and 0. Despite the function having a different shape, *logistic regression* is still considered a linear model. A *logistic regression*, for a binary outcome, calculates the probability for one of the outcomes $p(default)$, and has the formula:

$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$

This model is fit using *maximum likelihood*, which is basically fitting a bunch of S-shaped lines and calculating their probabilities given the data. The line with *maximum likelihood* is then picked.

In logistic regression the y-axis is transformed from *probability* to *log(odds)*, such that its y-axis can go from -infinity to +infinity, just like that of linear regression.

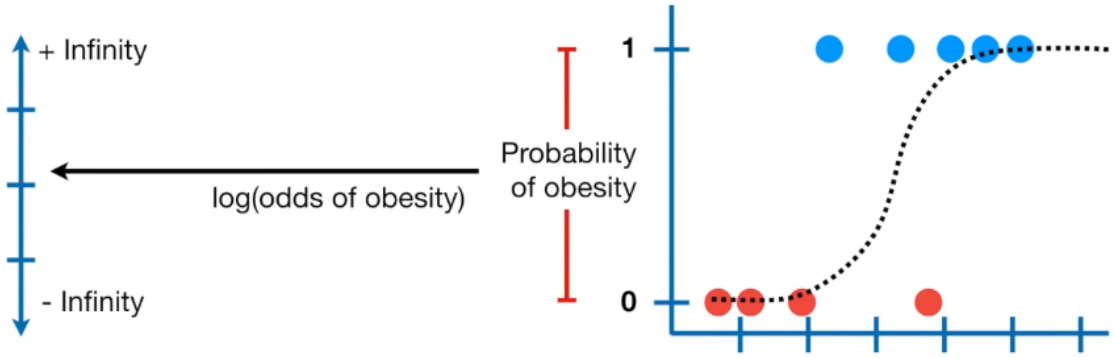


Figure 2: Probability vs Log Odds

The translation between *probability* and *log odds* can see for a sample in the table inderneath.

Propability	Odds	Log.Odds
0.1	0.1111111	-2.1972246
0.2	0.2500000	-1.3862944
0.3	0.4285714	-0.8472979
0.4	0.6666667	-0.4054651
0.5	1.0000000	0.0000000
0.6	1.5000000	0.4054651
0.7	2.3333333	0.8472979
0.8	4.0000000	1.3862944
0.9	9.0000000	2.1972246

1.1 Coefficients

The coefficients you get when doing logistic regression are for a linear line in a coordinate system where the y-axis is measured in *log(odds)*.

You get the following statistics:

- Intercept: Self-explanatory
- Slope: Self-explanatory
- Standard Error: How much one expects the value to vary from the real value

- Z-value: The estimated coefficient divided by the standard error, thus the number of standard deviations the coefficient is away from zero.
- p-value: How often you would expect to see this by random

1.2 Maximum Likelihood

Here we examine how we fit a line in a logistic regression. We do the following:

1. Project the original data points onto the candidate line
2. Transform the log(odds) into probabilities
3. Multiply all probabilities (p for target, 1-p for not target)
4. Repeat 1-3 for several lines, and find the line with the maximum likelihood.

1.3 R^2 and p-values

There is no standard way to calculate R^2 and p-values for logistic regressions, and there are more that ten different ways. It is normal just to do the same as what other people do within your field. We use *McFadden's Pseudo R^2*, which is similar to how R^2 is calculated for normal linear models:

1. For the fitted line project the original data points
2. Transform the log(odds) into probabilities
3. Calculate the *Log Likelihood*
4. Calculate the log(odds) for just the y-values, and draw a line
5. Project the data points onto the line
6. Transform the log(odds) into probabilities
7. Calculate the *Log Likelihood*
8. Calculate $R^2 = \frac{LL(overall\ probability) - LL(fit)}{LL(overall\ probability)}$

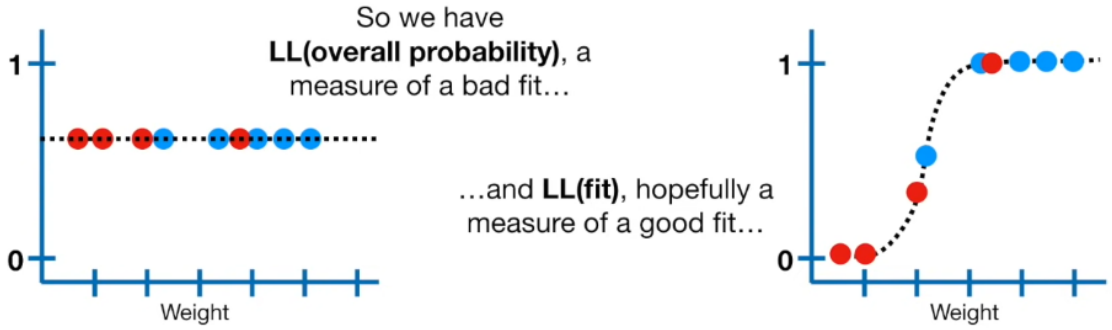


Figure 3: Logistic Regression R^2

https://www.youtube.com/watch?v=xxFYro8QuXA&list=PLblh5JKOoLUKxzEP5HA2d-Li7IJkHfXSe&index=4 09:36

Unlike linear regression, we can't easily compare the complicated model to the simple model e.g.:

Obesity predicted by **Weight + Genotype**
vs
Obesity predicted by **Weight + Genotype + Age**

STATQUEST: https://www.youtube.com/watch?v=yIYKR4sgzI8&list=PLblh5JKOoLUKxzEP5HA2d-Li7IJkHfXSe

X. Chapter Exercises & Answers

As most exercises are redundant for my brush up on statistical methods, I will only be solving select exercises. Great answers by Liam Morgan to all of the exercises can be found at RPubs.

X.1 Logistic Regression

Here we download the **Heart Disease Dataset** to classify if a person is healthy or unhealthy.

```
# Libraries
library(tidyverse)
library(cowplot)

# Downloading data
url <- "http://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/processed.cleveland.data
data <- read.csv(url, header=FALSE)

# Adding column names
colnames(data) <- c("age", "sex", "cp", "trestbps", "chol", "fbs", "restcg", "thalach", "exang", "oldpea

# Fixing column types
data[data == "?"] <- NA
data$sex <- ifelse(data$sex == 0, "F", "M")
data$ca <- as.integer(data$ca)
data$thal <- as.integer(data$thal)
data$hd <- ifelse(data$hd == 0, "Healthy", "Unhealthy")
factor_columns <- c("sex","cp", "fbs", "restcg", "exang", "slope", "ca", "thal", "hd")
data[factor_columns] <- lapply(data[factor_columns], factor)

# Removing samples with missing data
data <- data[!(is.na(data$ca) | is.na(data$thal)),]
```

Now we will make increasingly more complex models predicting health

```
# Making a logistic regression model
model <- glm(hd ~ sex, data=data, family = "binomial")
summary(model)
```

```
##
## Call:
## glm(formula = hd ~ sex, family = "binomial", data = data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.0438     0.2326  -4.488 7.18e-06 ***
## sexM          1.2737     0.2725   4.674 2.95e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 409.95  on 296  degrees of freedom
## Residual deviance: 386.12  on 295  degrees of freedom
## AIC: 390.12
##
## Number of Fisher Scoring iterations: 4
```

The output equations is:

$hearth\ disease = -1.0438 + IsMale * 1.2737$

The output value of the equation is the *log odds* that a person has a heart disease. Translation between probability, odds and log odds are best explained in this table.

Propability	Odds	Log.Odds
0.1	0.1111111	-2.1972246
0.2	0.2500000	-1.3862944
0.3	0.4285714	-0.8472979
0.4	0.6666667	-0.4054651
0.5	1.0000000	0.0000000
0.6	1.5000000	0.4054651
0.7	2.3333333	0.8472979
0.8	4.0000000	1.3862944
0.9	9.0000000	2.1972246

So the results are interpreted as if your *log odds* for heart disease will increase by 1.2737 if you are a male. The *standard error* and *z-value* are for calculating the *Wald's test*, while the p-value states its significance.

Now it is time for a mode fancy models with all the variables.

```
# Making a logistic regression model
model <- glm(hd ~ ., data=data, family = "binomial")
summary(model)
```

```
##
## Call:
## glm(formula = hd ~ ., family = "binomial", data = data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.253978   2.960399  -2.113 0.034640 *
## age         -0.023508   0.025122  -0.936 0.349402
## sexM         1.670152   0.552486   3.023 0.002503 **
## cp2          1.448396   0.809136   1.790 0.073446 .
## cp3          0.393353   0.700338   0.562 0.574347
## cp4          2.373287   0.709094   3.347 0.000817 ***
## trestbps     0.027720   0.011748   2.359 0.018300 *
## chol         0.004445   0.004091   1.087 0.277253
## fbs1        -0.574079   0.592539  -0.969 0.332622
## restcg1      1.000887   2.638393   0.379 0.704424
## restcg2      0.486408   0.396327   1.227 0.219713
## thalach     -0.019695   0.011717  -1.681 0.092781 .
## exang1       0.653306   0.447445   1.460 0.144267
## oldpeak      0.390679   0.239173   1.633 0.102373
```

```
## slope2       1.302289   0.486197   2.679 0.007395 **
## slope3       0.606760   0.939324   0.646 0.518309
## ca1          2.237444   0.514770   4.346 1.38e-05 ***
## ca2          3.271852   0.785123   4.167 3.08e-05 ***
## ca3          2.188715   0.928644   2.357 0.018428 *
## thal6       -0.168439   0.810310  -0.208 0.835331
## thal7        1.433319   0.440567   3.253 0.001141 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 409.95  on 296  degrees of freedom
## Residual deviance: 183.10  on 276  degrees of freedom
## AIC: 225.1
##
## Number of Fisher Scoring iterations: 6
```

```
# Calculating McFadden's Pseudo R^2
ll.null <- model$null.deviance/-2
ll.proposed <- model$deviance/-2
(ll.null-ll.proposed)/ll.null
```

```
## [1] 0.5533531
```

```
# Calculating p-value using a chi-squared distribution
1 - pchisq(2*(ll.proposed - ll.null), df=(length(model$coefficients)-1))
```

```
## [1] 0
```

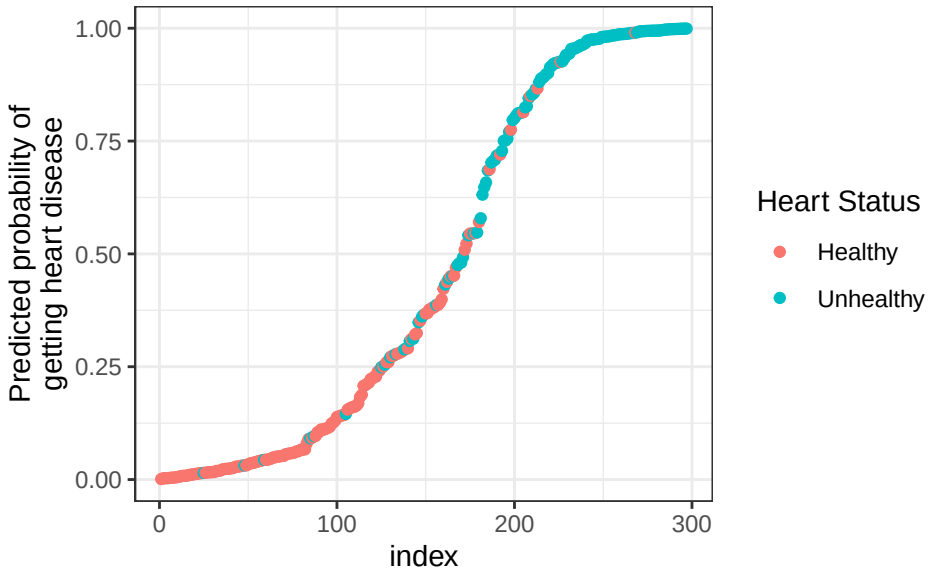
The *McFaddan Pseudo* R^2 can be interpreted as the overall effect size, and its p-value is significant as it if very close to zero. Now let's draw a graph with the predictions.

```
# Making predictions
predicted.data <- data.frame(
  probability.of.hd=model$fitted.values,
  hd=data$hd
)

# Ordering data
predicted.data <- predicted.data[order(predicted.data$probability.of.hd, decreasing = F),]

# Adding index
predicted.data$index <- seq(1, nrow(predicted.data))

# Plotting data
ggplot(predicted.data, aes(index, probability.of.hd, color=hd)) +
  geom_point() +
  ylab("Predicted probability of\ngetting heart disease") +
  labs(color='Heart Status') +
  theme_bw()
```




```
# Libraries
library(tidyverse)
```

NEXT: “Odds and Log(Odds) clearly explained”, “Odds Ratios and Log(Odds Ratios) clearly explained”, “Saturated Models and Deviance Statistics”