Logistic Regression

Wai Yu Amanda Ng - 1008217350

October 8 2022

Abstract

This report will give an introduction to logistic regression, including how should logistic regression model look like, the restrictions of logistic regression and R codes to produce, test and visualize logistic regression models. In this report, we will focus on exploring binary simple logistic regression model. The report is meant for readers with basic knowledge about statistical prediction models such as linear regression model.

Keywords:

Logistic Regression, Classification, Continuous variable, Discrete variable, Predictor, Response variable

Definitions:

Continuous variable can take any value within a range (eg. height)

Discrete variable can only take some value in a range (eg. number of students in STA299; it does not make sense to say there are 34.7 students in the class)

Predictor variable is the name given to an independent variable used in regression analyses. The predictor variable provides information on an associated dependent variable regarding a particular outcome.

Response variable is the name given to an dependent variable used in regression analyses. It is a measure of the response of an item in the data set to the treatment provided with respect to the predictor variable.

An observation is a fact or figure we collect about a given variable. It can be expressed as a number or as a quality.

Review of linear regression

Linear regression is a statistical method where a linear model is built of which we can analyze if a **linear** relationship between a predictor variable x and a single response variable y is present. If a linear relationship exists, it will be of positive association where values of x tend to increase as that of y increases or, vice versa, negative association. This model assists us to understand variation in x and predict y by some calculation through a linear combination of the predictor x which is represented by a linear regression equation.

A simple fitted linear regression equation will look as follow:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

where

- x is the predictor
- $\hat{y}$ is the predicted value (estimated average value of y when predictor $= x$)
- β_0 is the intercept parameter (average of y when $x = 0$)
- β_1 is slope parameter (average change of y when x changes by 1 unit)

A multiple fitted linear regression equation contains more than one predictor(x).

For example, the following fitted equation consists of two predictors and the graph will consist 2 parallel lines.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2$$

We can also build multiple linear regression equations which gives non-parallel lines.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_1 x_2$$

However, linear regression restricts the response variable (y) to be a continuous numerical variable.To build regression equations for discrete response variable (y), especially when y is a response of consists of 2 levels, such as answers to a yes-no question, logistic regression model may be helpful.

Introduction to Logistic Regression

1. When can we use logistic regression?

Logistic regression can used to model and solve **binary classification problems**. That is, given a predictor x (it can be continuous or discrete), we can use logistic regression model to predict the value of a response variable y which consists of only two classes.

2. How does a logistic regression model look like?

A logistic regression model does not give a straight line like linear regression model. Instead, it fits a **'S' shaped 'logistic function'** to the data and the curve goes from 0 to 1.

3. Logistic regression model equation and explanation

A logistic regression model can be given by a function:

$$p(x) = 1/(1 + e^{-(\beta_0+\beta_1 x)})$$

where β_0 is the y-intercept of the S-curve and $beta_1$ is the rate parameter.

Let the two possible outcomes of the binary response variable y be {1,0}. Note that we put $p(x)$ instead of y as the output of the function because we restricted y to be either 1 or 0 but the function gives continuous value, representing the probability of $y = 1$.

For any point on the S-curve, its y-coordinate represents the probability of $y = 1$ given such x. Hence, by definition, probability of $y = 1$ can vary between 0 (certainly $y = 0$) and 1 (certainly $y = 1$).

How to use a logistic regression model?

Although the logistic regression model gives us probabilities of y being one of the possible outcomes (whichever we define as '1' on the plot) for a given x, we usually use it for classification only. Generally, when the possibility obtained for a x is less than 50%, we will classify its outcome as $y = 0$. Conversely, if the possibility is greater than 50%, we will predict its outcome as $y = 1$

A mini example

Scenario: We want to predict if a person is suffering from obesity based on their BMI.

Here, we have BMI as the indicator x and the response variable y consists of two possible outcome {obese, not obese}.

We can label the outcome 'obese' as 1 and 'not obese' as 0.

Given a built logistic regression model S-curve, We draw a horizontal line $y = 0.5$ on the graph. Any points on the curve which are below $y = 1$ has a probability of less than 50%, i.e. $P(yes) < 0.5$. Similarly, those above $y = 0.5$ has a probability of greater than 50%, i.e. $P(yes) > 0.5$.

Assume $y = 0.5$ cut the curve at the point $(30.0, 0.5)$.

1. Let Amanda's BMI be 20.0, which is smaller than 30.0, the possibility of her being obese is less than 50%, we can classify her as not obese.
2. Let Brian's BMI be 31.0, which is higher than 30.0, the possibility of him being obese is greater than 50%, we can classify him as obese.

How to build a simple binary logistic regression model

1. Identify variables

Firstly, we should identify the predictor(s) x and response variable y in our problem. It is important to make sure that our response variable y is:

- a single output
- binary, meaning that it only has two possible outcomes

2. Data

2a. Data collection process

Secondly, we have to collect a data set which serves as the foundation of the development of the logistic model. It is always better to collect as much as data as possible to increase the model's accuracy and minimize biases or error created in data collection.

Below are some cautions to be aware of during data collection: * Remove potential outliers * Make sure that the predictor variables are normally distributed. * Remove highly correlated predictors to minimize over-fitting. The presence of highly correlated predictors might lead to an unstable model solution.

We can then input the data set to R as a data frame.

2b. Data Summary

Let's consider the scenario where we want to predict people's status of diabetes-positive based on their plasma glucose concentration.

The starting data set *'PimaIndiansDiabetes2'* of size 768 is adapted from the R package *'mlbench'*. Our predictor x is *'plasma glucose concentration'* and response variable y is binary, namely {positive, negative}.

The cleaning process involves removing observations in which *'plasma glucose concentration'* is not available. Eventually, there remains 763 data that are suitable for modeling.

```
library('mlbench')
# Load data set
data(PimaIndiansDiabetes2)

# Cleaning process
data <- PimaIndiansDiabetes2 %>%
   filter(!(is.na(glucose)))

# Size of our sample set for modelling
n <- nrow(data)
```

To prepare for modeling, we will divide the sample data set randomly into two parts : training set and testing set.

This is to ensure different observations are used for fitting and testing the model we are going to build.

```
set.seed(123)

# Assigning id number to each data
data1 <- data %>%
  rowid_to_column()

# Choose randomly 70% of the sample data set as training set
# and remaining 30% as testing set
training_indices <- sample(1:n, size = round(0.7*n))
```

```r
train <- data1 %>%
  filter(rowid%in%training_indices)

test <- data1 %>%
  filter(!(rowid%in%training_indices))
```

In the above R code, we save the 70% training data set under the variable named 'train' and the other 30% testing data set under the variable named 'test'.

3. Model

3a. Model building

The function **glm()** is very helpful in building logistic regression model. The first section of the inputs is the formula y~x, representing x predicts y. The second section describes the data set to be fitted into the model. We need to specify the option family = binomial, which tells to R that we want to fit logistic regression.

```r
# Use the funtion glm() to build the model
model <- glm(diabetes ~ glucose, data = train, family = binomial)

# Coefficients of the model
summary(model)$coef
```

```
##                Estimate Std. Error   z value     Pr(>|z|)
## (Intercept) -5.69437665 0.51996286 -10.95151 6.535157e-28
## glucose      0.04068651 0.00401347  10.13749 3.766575e-24
```

Hence, our simple logistic regression model is:

$$p(x) = 1/(1 + e^{-(-5.6944+0.040687x)})$$

3b. Model testing

As mentioned above, we retained 30% of the sample data set for testing on the model we just built. Now, we want to measure how accurately the regression model can predict the testing set.

```r
# Make probability predictions on testing data set
probabilities <- model %>% predict(test, type = "response")

# Classification based on probabilities
predicted.classes <- ifelse(probabilities > 0.5, "pos", "neg")
predictions <-data.frame(predicted.classes)
```

The predicted probabilities (i.e. P(diabetes-positive)) of observations in the testing data set are saved as a vector under the variable 'probabilities'. Next, we classify the observations as {positive} or {negative} based on the probabilities and save them as a data frame under the variable 'predictions'.

4. Measuring accuracy of the model

```
# Compare diabetes-positive status stated in the observations
# of testing data set and that of predicted by the model
data.frame(TorF = predictions$predicted.classes == test$diabetes) %>%
  mutate(match = case_when(TorF == TRUE ~ 'match', TorF == FALSE ~ 'no match'))%>%
  group_by(match)%>%
  summarize(n())
```

```
## # A tibble: 2 x 2
##   match    `n()`
##   <chr>    <int>
## 1 match      175
## 2 no match    54
```

Category	Count
match	175
no match	54

```
# Proportion of correctly predicted observations in test data set
mean(predicted.classes == test$diabetes)
```

```
## [1] 0.7641921
```

It shows that the model predicts 175 out of 229 test set's the status of diabetes-positive correctly (i.e. there is a match between the prediction and real status), which gives around 76.4% of accuracy.

McFadden's pseudo R-squared

We indeed have a more formal way to measure the credibility of the logistic regression model, which is the McFadden's pseudo R^2.

Logistic regression models are fitted using the method of maximum likelihood – i.e. the parameter estimates are those values which maximize the likelihood of the data which have been observed.

The equation of this R^2 is different from that of linear regression model. It is given by:

$$R^2_{McFadden} = 1 - (Ln(M_c)/Ln(M_{null}))$$

where M_c denotes the (maximized) likelihood value from the current fitted model, and M_{null} denotes the corresponding value from the null model which does not consist any predictor.

A likelihood function calculates the probability of observing the outcome given the input data and the model. This function can then be optimized to find the set of parameters that results in the largest sum likelihood over the training data set.

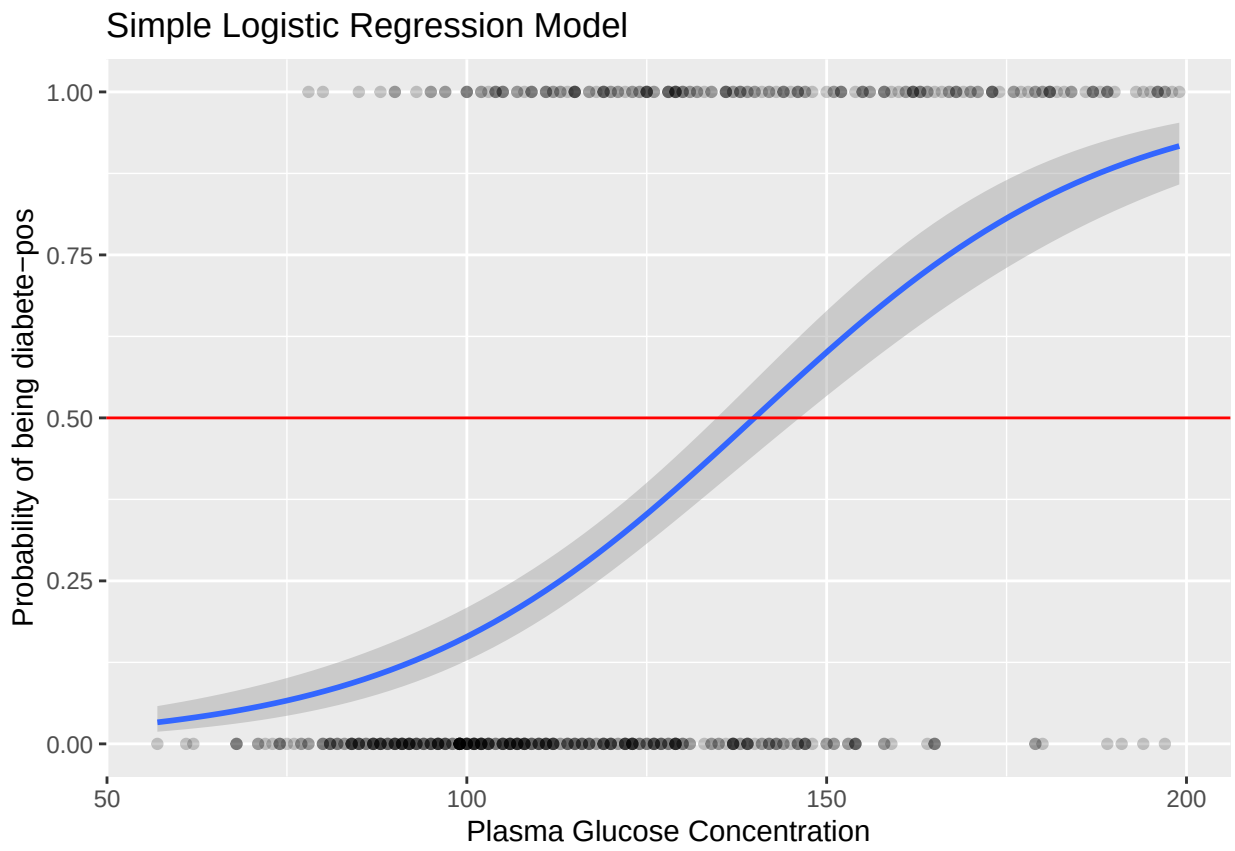
The following R-code helps us compute $R^2_{McFadden}$ nicely. We only need to input the model variable into the function summary ().

```
with(summary(model), 1 - deviance/null.deviance)
```

```
## [1] 0.2047022
```

In general, if $R^2_{McFadden}$ is between 0.2 and 0.4, we consider the fitted model as a good model. Hence, this model is acceptable to be used for predictions.

5. Graph of model



The blue S-curve is the fitted logistic regression model. The red horizontal line at y=0.5 is the cut-off line. Any point on the curve which is below the red line is considered as diabetes-negative and those above the red line are considered as diabetes-positive.

6. Results

According to the graph in part 5, if a person's plasma glucose concentration is below 139.96, they are considered as diabetes negative. Conversely, if the plasma glucose concentration is above 139.96, they are considered as diabetes positive.

Conclusions

Logistic regression is a useful tool for predicting binary outcomes. Similar to linear regression, there is an explicit equation for the model and R^2 (although with a different formula) to measure the model's accuracy.

Weaknesses

Below are some disadvantages of logistic regression:

- It assumes linearity between the dependent variable and the independent variables.
- If the number of observations is lesser than the number of predictors, Logistic Regression should not be used, otherwise, it may lead to over-fitting.
- It can only be applied to predictions on binary response variable.

Next Steps

We have only showcased steps and R codes in building a *simple binary* logistic regression model. However, multiple binary logistic regression can also be produced by including more predictors and maintaining a single binary response variable. Moreover, models to predict responses of 3 or more levels can also be discussed in the topic of Nominal Logistic Regression (where ranking does not matter) and Ordinal Logistic Regression (where ranking matters).

References

Brownlee, J. (2019, October 27). A gentle introduction to logistic regression with maximum likelihood estimation. Machine Learning Mastery. Retrieved October 6, 2022, from https://machinelearningmastery.com/logistic-regression-with-maximum-likelihood-estimation/

Kassambara. (2018, March 11). Logistic Regression Essentials in R. STHDA. Retrieved October 6, 2022, from http://www.sthda.com/english/articles/36-classification-methods-essentials/151-logistic-regression-essentials-in-r/

Logistic regression: A simple beginner's guide in 4 steps. Jigsaw Academy. (2022, July 6). Retrieved October 6, 2022, from https://www.jigsawacademy.com/blogs/data-science/logistic-regression/

Pimaindiansdiabetes: Pima indians diabetes database. RDocumentation. (n.d.). Retrieved October 6, 2022, from https://www.rdocumentation.org/packages/mlbench/versions/2.1-3/topics/PimaIndiansDiabetes

Prabhakaran, S. (2022, March 9). Logistic regression - a complete tutorial with examples in R. Machine Learning Plus. Retrieved October 6, 2022, from https://www.machinelearningplus.com/machine-learning/logistic-regression-tutorial-examples-r/

Starmer, J. (2018, March 5). StatQuest: Logistic regression. YouTube. Retrieved October 6, 2022, from https://www.youtube.com/watch?v=yIYKR4sgzI8

Appendix

A1: Ethics Statement

The open data set *PimaIndiansDiabetes2* contributed by UCI Repository Of Machine Learning Databases and owned by National Institute of Diabetes and Digestive and Kidney Diseases is used in the report.

A2: Materials

R code of the graph in section *5. Graph of model*

```
train %>%
  mutate(prob = ifelse(diabetes == "pos", 1, 0)) %>%
  ggplot(aes(glucose, prob)) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "glm", method.args = list(family = "binomial")) +
  geom_hline(yintercept=0.5, color = 'red')+
  labs(
    title = "Simple Logistic Regression Model",
    x = "Plasma Glucose Concentration",
    y = "Probability of being diabete-pos"
    )
```