

MA5810: Introduction to Data Mining

Week 3; Collaborate Session 1: Logistic Regression

Martha Cooper, PhD

JCU Masters of Data Science

2019-21-9 (updated: 2020-11-11)

Housekeeping

- Collaborate 1 = [Wednesdays 6-7pm](#) (Martha)
- Collaborate 2 = [Thursdays 7-8pm](#) (Hongbin)

For my Collaborate Sessions, you can get the [slides & R code](#) for each week here:

<https://github.com/MarthaCooper/MA5810>



Assignment 1 Q1

Explain **why** you chose the algorithm based on:

- 1) The algorithm assumptions;
- 2) How your data relates to those assumptions.

The question is **not** about calculating a confusion matrix or ROC. You do not need to do that!

Subject: MA5810 Intro to Data Mining

MA5810 Learning Outcomes

1. Overview of Data Mining and Examples
2. Unsupervised data mining methods e.g. clustering and outlier detection;
3. Unsupervised and supervised techniques for dimensionality reduction;
4. Supervised data mining methods for pattern classification (Today = Logistic Regression);
5. Apply these concepts to real data sets using R (Today).

Today's Goals

- Understand the background behind Logistic Regression
- Apply Logistic Regressions to real datasets using R
- Understand the pros and cons of Logistic Regression

Linear Regression Review

- The simple linear regression model is:

$$Y = \beta_0 + \beta_1 X + \epsilon$$

Where

- Y is the dependent variable
- X is the independent variable
- β_0 is the intercept (Y when $X = 0$)
- β_1 is the slope of the regression line
- ϵ is the error term

Multiple Linear Regression Review

- Multiple regression with k independent variables

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_k X_k + \varepsilon$$

- When interpreting one of the slopes in multiple regression model, we should take into account the effect of the other variables
- For instance, β_1 represents the change in Y per 1 unit change in X_1 , holding other variables (X_2, \dots, X_k) constant

Generalised Linear Models & Classification

- **GLM**: Appropriate when Y isn't normally distributed but is in the exponential family of distributions
- In classification where Y is **binomial** (or multinomial)
- Given these features, does this sample belong to class A or B?

cancer $\in \{yes, no\}$

credit card $\in \{default, not\ default\}$

win $\in \{yes, no\}$

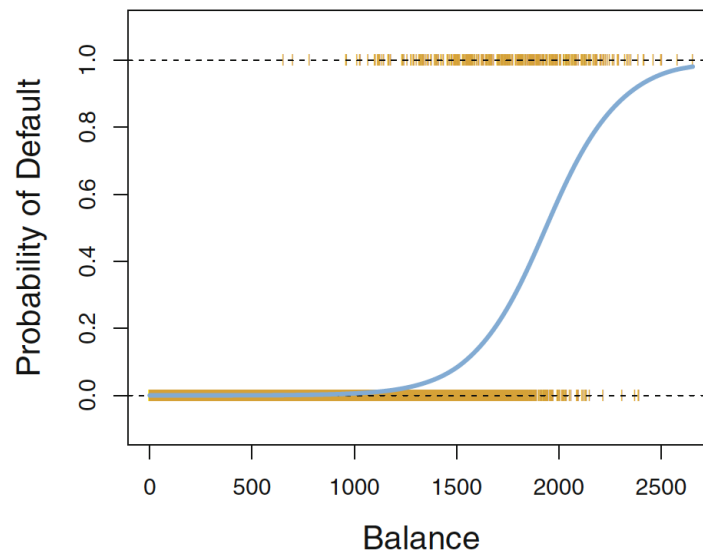
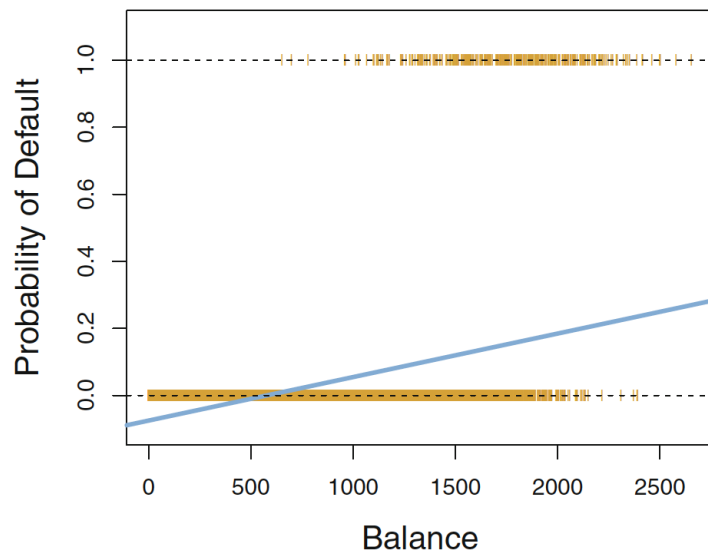
drug $\in \{survived, not\ survived\}$

Logistic Regression

- Binomial family Generalised Linear Model
- Models the probability that Y belongs to a particular category

Logistic Regression

- Binomial family Generalised Linear Model
- Models the probability that a subject belongs to a particular category

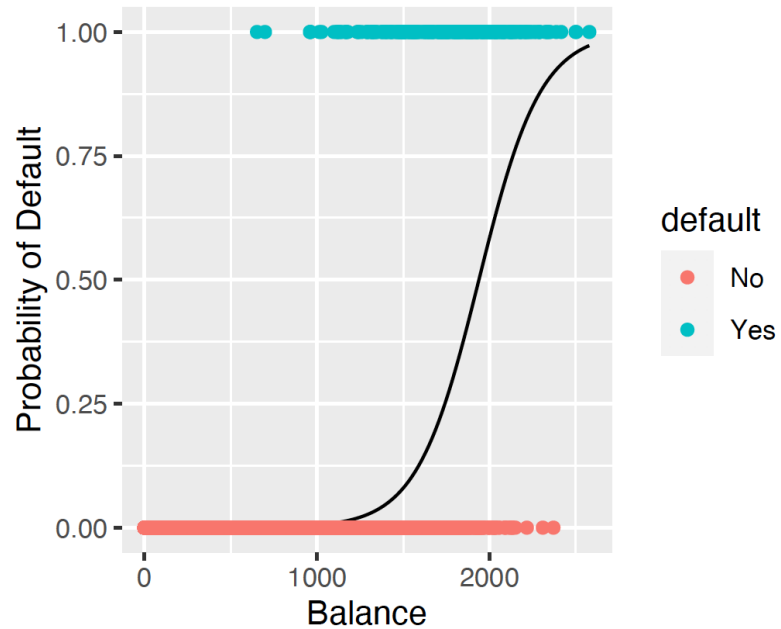


Problems with Linear Regression for Classification

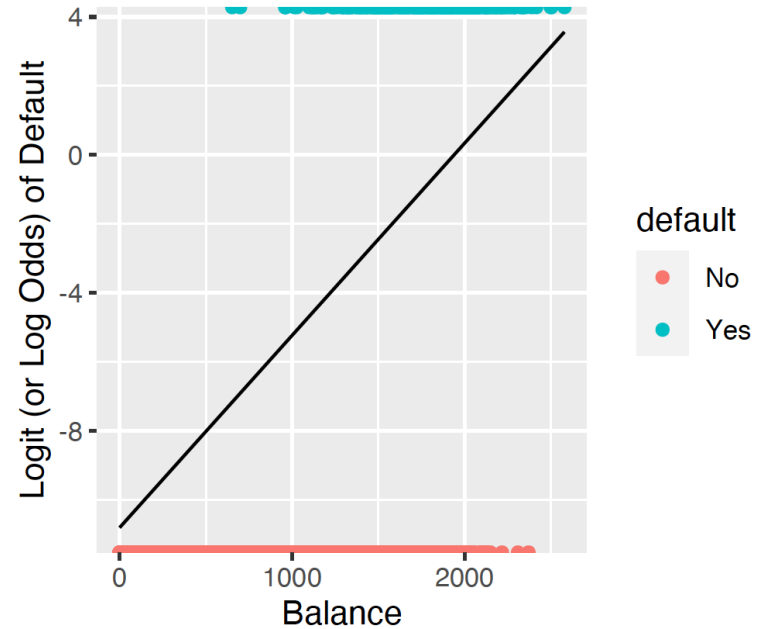
- Some values are outside $[0,1]$
- For multinomial classification, the order and interval between classes would be considered important and meaningful

The Logistic Model

Probability



Logit or Log Odds



The Logistic Model

Let $P(Y = 1|X)$ be the probability that $Y = 1$ given $X = (X_1, \dots, X_k)$

Probability

$$P(Y = 1|X_1, \dots, X_k) = \frac{e^{\beta_0 + \beta_1 X_1 + \dots + \beta_k X_k}}{1 + e^{\beta_0 + \beta_1 X_1 + \dots + \beta_k X_k}}$$

- where e is the Euler's number.
- This function means that $0 \leq P(Y = 1|X) \leq 1$

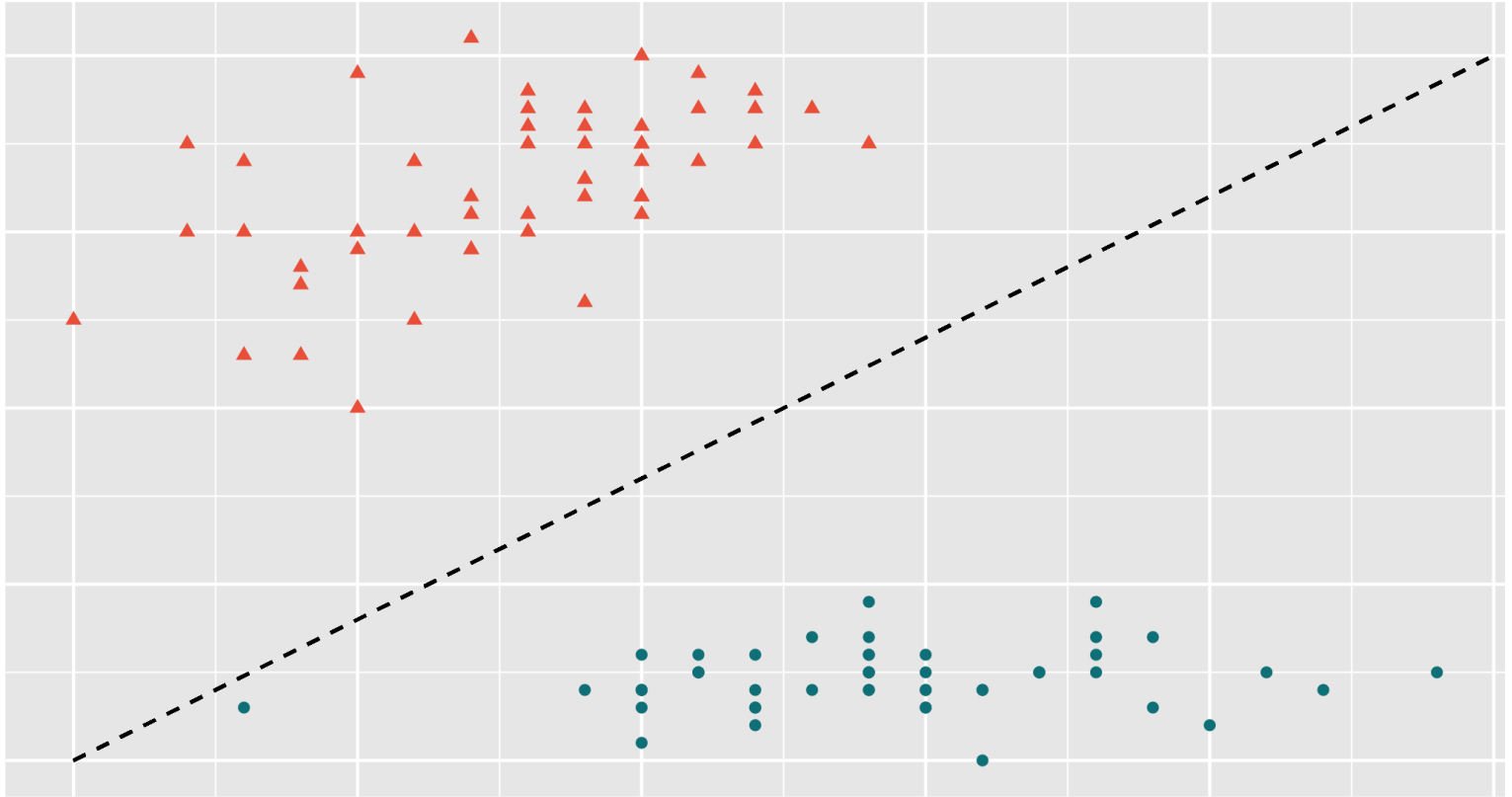
Logit (Log Odds)

$$\log\left(\frac{P(Y = 1|X_1, \dots, X_k)}{1 - P(Y = 1|X_1, \dots, X_k)}\right) = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k$$

- Where \log is the natural log, \log_e
- Interpretation: β_1 represents the change in **log odds** of Y per 1 unit change in X_1 , holding other variables (X_2, \dots, X_k) constant

Estimating the coefficients

Maximum Likelihood



Logistic Regression Pros and Cons

Pros

- Identify which features are important for classification
- Interpret how important each feature is for classification

Cons

- Doesn't perform well if the decision boundary isn't linear
- Two groups (although extensions make more possible)

Logistic Regression in R

```
#load data
library(ISLR, warn.conflicts = F, quietly = T) #for data
library(caret, warn.conflicts = F, quietly = T) #for splitting the data
library(dplyr, warn.conflicts = F, quietly = T) #for piping

data("Default") #credit card default data from ISLR
str(Default)

#split into training (80%) and test
split <- createDataPartition(Default$default, p = 0.8, list = F)

train <- Default[split, ]
test <- Default[-split, ]

c(nrow(train), nrow(test)) # print number of observations in test vs. train

table(train$default) %>% prop.table() # Proportions of people that default

#Train the model to predict the likelihood of default status based on credit balance
def_logmod1 <- glm(default ~ balance, data = train, family = "binomial")
```

Interpreting the coefficients

```
summary(def_logmod1)$coef #interpret the coefficients
```

##		Estimate	Std. Error	z value	Pr(> z)
##	(Intercept)	-10.750888865	0.4111675862	-26.14722	1.059975e-150
##	balance	0.005563878	0.0002505912	22.20301	3.212218e-109

- β_0 : Log odds
- β_1 : Log odds ratio

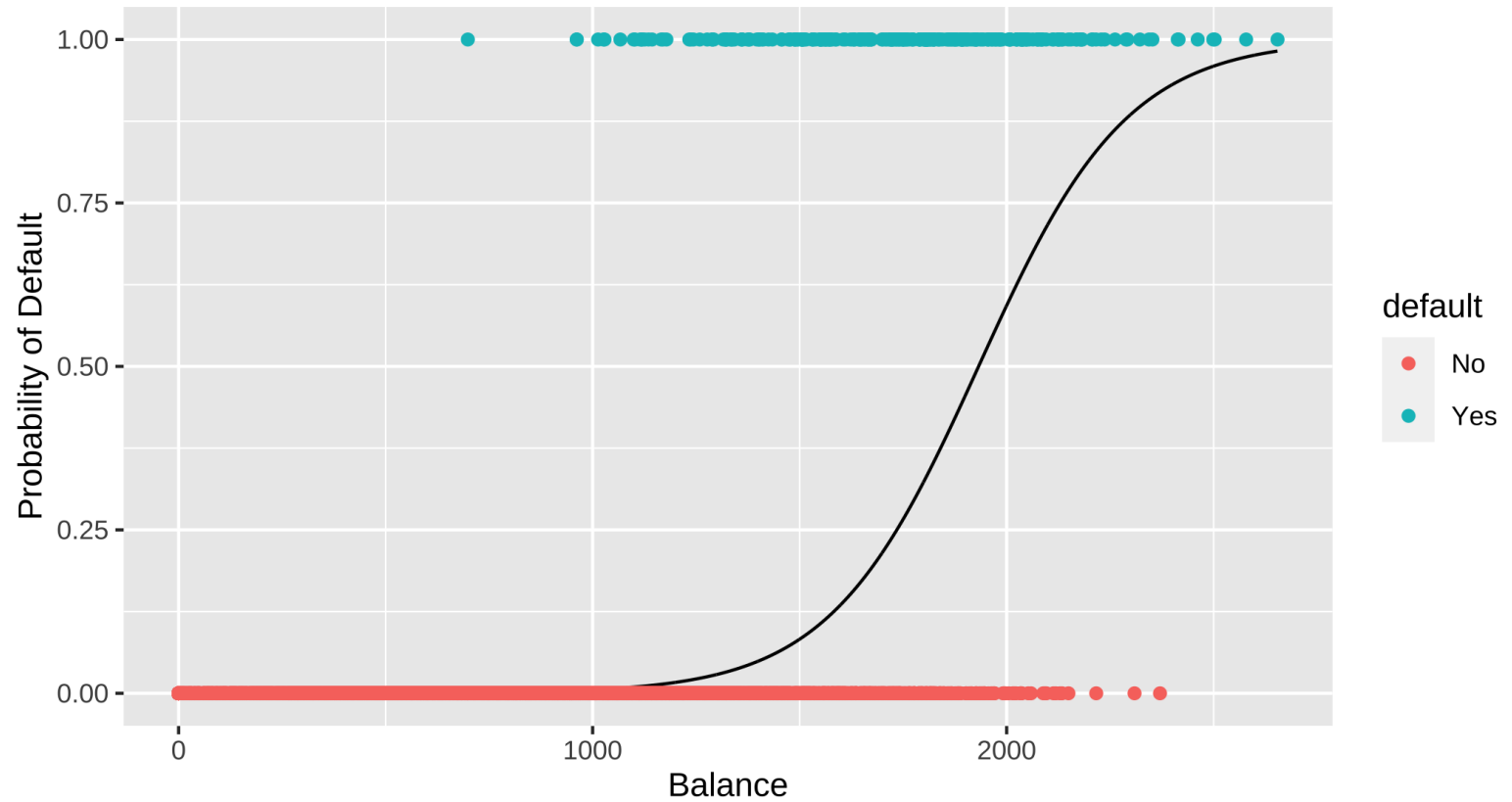
Making predictions in R

- make predictions based on **training** data

```
l odds <- predict(def_logmod1, type = "link") #log odds
preds_lodds = ifelse(l odds > 0, "Yes", "No") #using log odds
confusionMatrix(as.factor(preds_lodds), train$default) #confusion matrix
```


Plot the model

- We are aiming for a plot like this:



Plot the model

```
def_logmod1$coef #look at coefs
#save coefficients
b0 <- def_logmod1$coef[1] #beta0
b1 <- def_logmod1$coef[2] #beta1

#calculate probabilities
x_range <- seq(from = min(train$balance), to = max(train$balance)) #range
#calculate the logits
default_logits <- b0 + b1*x_range

#calculate probabilities to plot
default_probabilities <- exp(balance_logits)/(1 + exp(balance_logits))

probabilities_to_plot <- data.frame("balance" = x_range,
                                   "probability_of_default" = default_probabilities)

ggplot(probabilities_to_plot, aes(x = balance, y = probability_of_default)) +
  geom_line() + #plot model
  geom_point(data = train, aes(x = balance,
                              y = ifelse(default == "Yes", 1, 0),
                              colour = default)) + #add training data
  xlab("Balance")
```

Making predictions in R

- make predictions based on `test` data

```
test_lodds <- predict(def_logmod1, newdata = test, type = "link") #logit  
test_preds_lodds = ifelse(test_lodds > 0, "Yes", "No") #using logits  
confusionMatrix(as.factor(test_preds_lodds), test$default) #confusion matrix
```

Extra reading

- Chapter 4 of James *et al.*, [ISLR](#)
- Chapter 10 of David Dalpiaz, [R for Statistical Learning](#)

References

- James *et al.*, [ISLR](#)
- David Dalpiaz, [R for Statistical Learning](#)

Slides

- xaringhan, xaringantheme, remark.js, knitr, R Markdown