MA5810: Introduction to Data Mining

Week 3; Collaborate Session 1: Logistic Regression

Martha Cooper, PhD

JCU Masters of Data Science

2021-07-20

Housekeeping

- Collaborate 1 = Tuesday 6.45-8pm (Martha)
- Collaborate 2 = Thursday 6.45-8pm (Martina)

For my Collaborate Sessions, you can get the slides & R code for each week here:

https://github.com/MarthaCooper/MA5810

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

β_0 and β_1 are estimated by Ordinary Least Squares

- Minimizing the sum of the squared distances between the observed values, $\boldsymbol{y}$, and the fitted values, $\hat{\boldsymbol{y}}$. (*i.e.* residuals)

$$\sum_{i=1}^{n} (y_i - (\beta_0 - \beta_1 x_i))^2$$

Multiple Linear Regression Review

- Multiple regression with k independent variables

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

- 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 $(X_2, \ldots, X_k)$ 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?

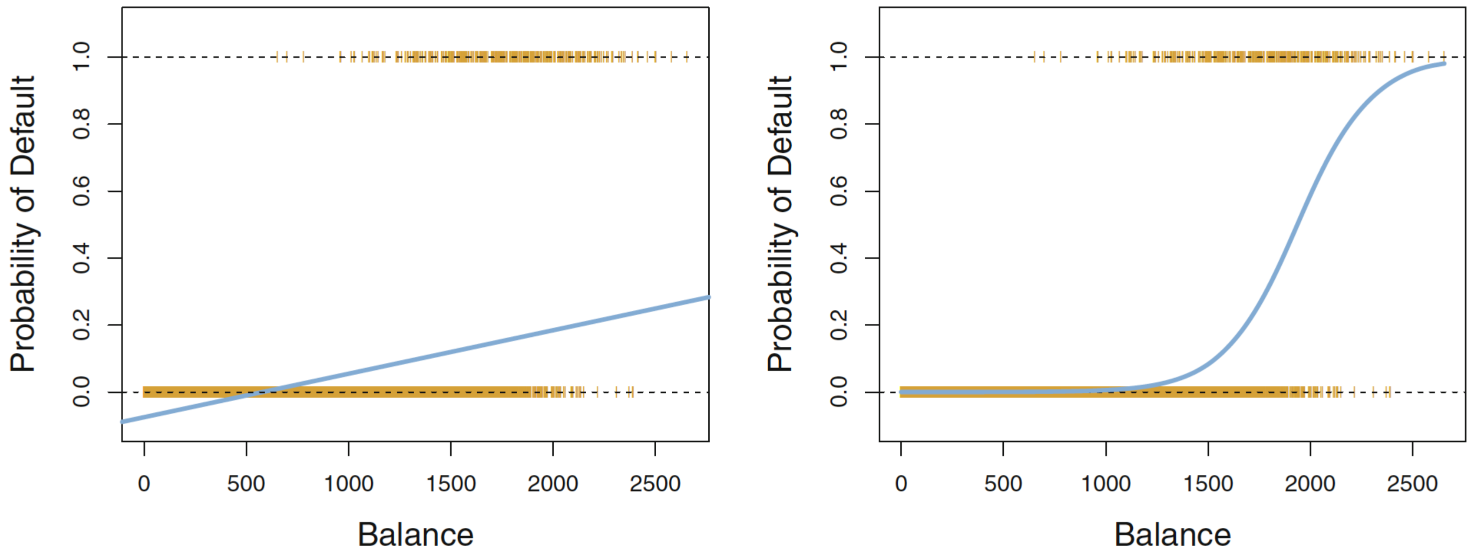
$$\begin{aligned} cancer &\in \{yes, no\} \\ credit\ card &\in \{defalt, not\ default\} \\ win\ &\in \{yes, no\} \\ drug\ &\in \{survived, not\ survived\} \end{aligned}$$

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

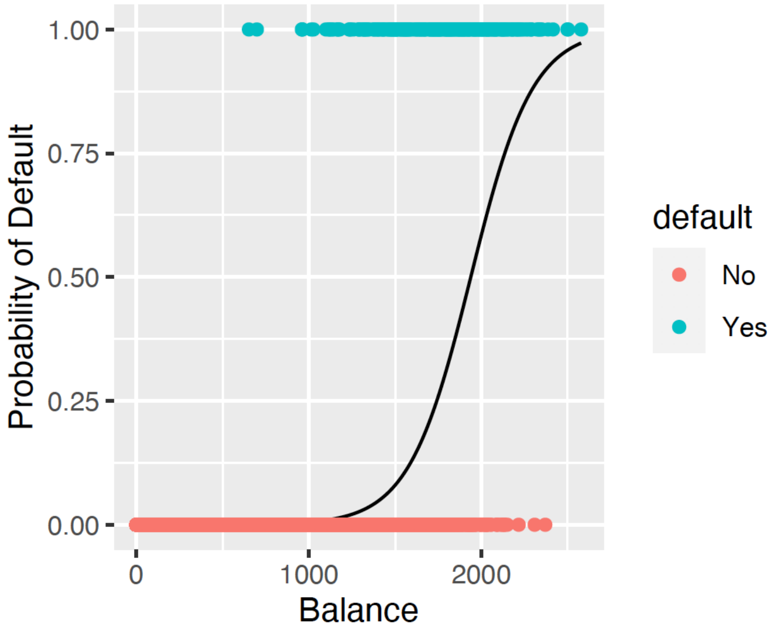


Problems with Linear Regression for Classification

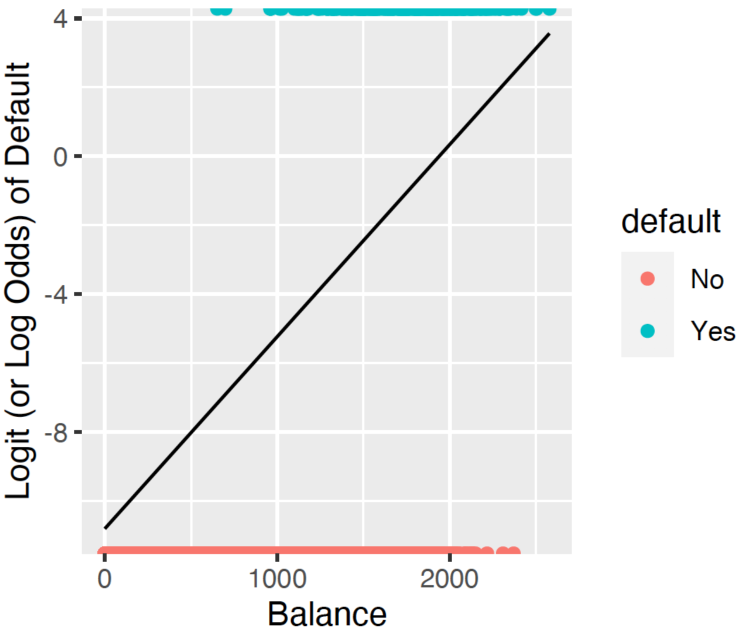
- Some values are outside [0,1]
- For multinomial classification, the order and interval between classes would be considered important and meaningful

The Logistic Model

Probability



Logit or Log Odds



The Logistic Model

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

Probability

$$P(Y=1|X_1,...,X_k)=\frac{e^{\beta_0+\beta_1X_1+...+\beta_kX_k}}{1+e^{\beta_0+\beta_1X_1+...+\beta_kX_k}}$$

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

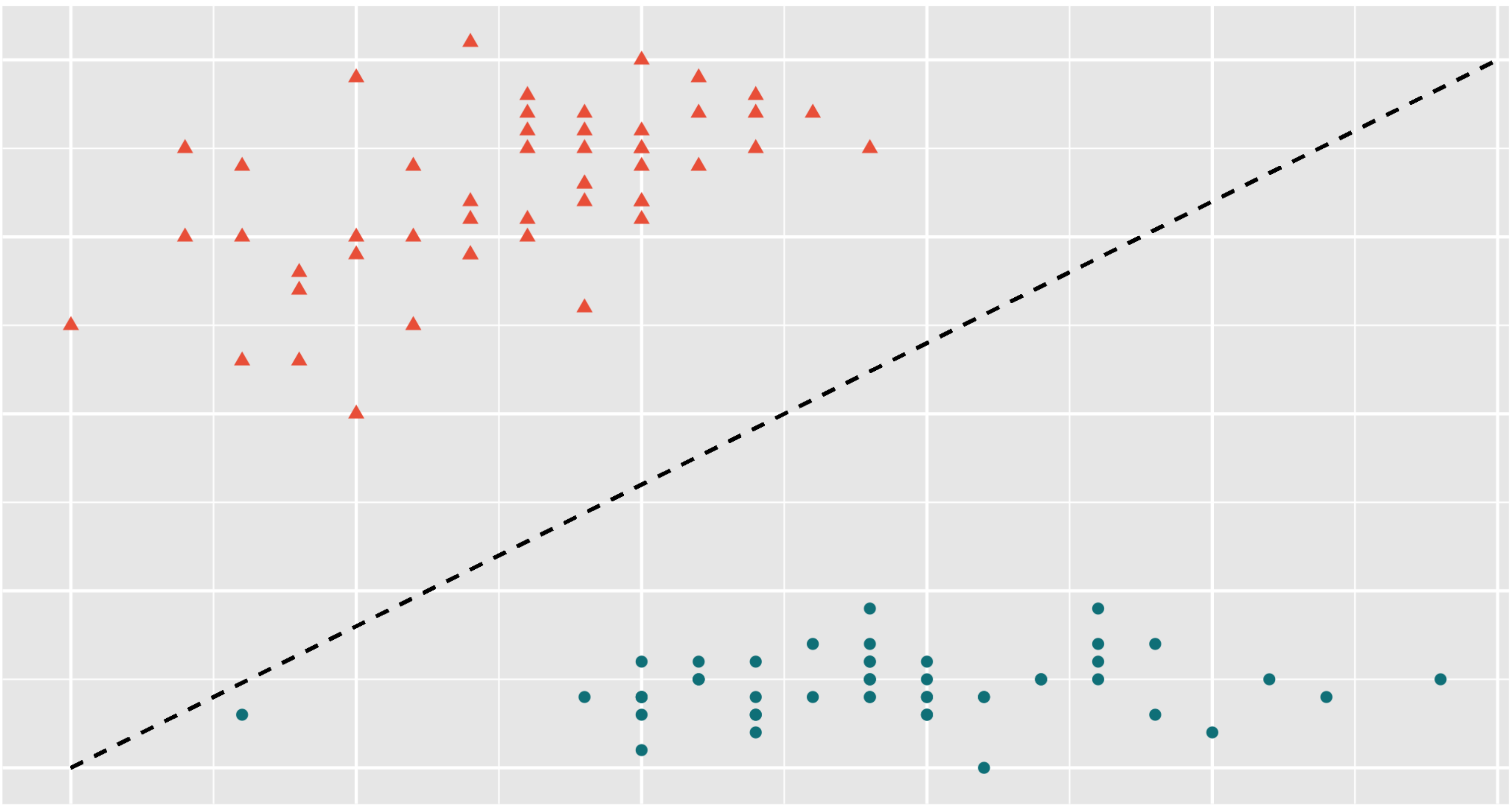
Logit (Log Odds)

$$\log(\frac{P(Y=1|X_1,...,X_k)}{1-P(Y=1|X_1,...,X_k)})=\beta_0+\beta_1X_1+...+\beta_kX_k$$

- 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 $(X_2,...,X_k)$ constant

Estimating the coefficients

Maximum Likelihood



- Estimate coefficients such that the predicted probability class membership is as close as possible to the individuals observed class

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
set.seed(123)
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. t

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

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

Interpretting the coefficients

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

```
##                  Estimate    Std. Error   z value      Pr(>|z|)
## (Intercept) -10.649065060 0.4030015555 -26.42438 7.190404e-154
## balance       0.005490249 0.0002457254  22.34303 1.411387e-110
```

- β_0 : the log odds of a person defaulting if they have a credit card balance of 0 dollars = -10.649
- β_1 : Log odds ratio -for a one unit increase in balance, the log odds of a person defaulting increases by 0.00549

```
exp(def_logmod1$coeff) #exponentiate the coefficients to see the odds ra
```

```
##  (Intercept)      balance
## 2.372301e-05 1.005505e+00
```

```
# odds of defaulting increase by 0.55% for every $1
```

Making predictions - theory

- Once we have the coefficients, we can calculate the probability of default for any credit card balance

```
summary(def_logmod1)$coef
```

```
##                  Estimate    Std. Error   z value      Pr(>|z|)
## (Intercept) -10.649065060 0.4030015555 -26.42438 7.190404e-154
## balance       0.005490249 0.0002457254  22.34303 1.411387e-110
```

- For example, if the balance is 1000 dollars

$$\frac{e^{-10.64+0.0054*1000}}{1+e^{-10.64+0.0054*1000}} = 0.0057 = 0.57\%$$

- For example, if the balance is 2000 dollars

$$\frac{e^{-10.6513+0.0057*2000}}{1+e^{-10.6513+0.0057*2000}} = 0.57 = 57\%$$

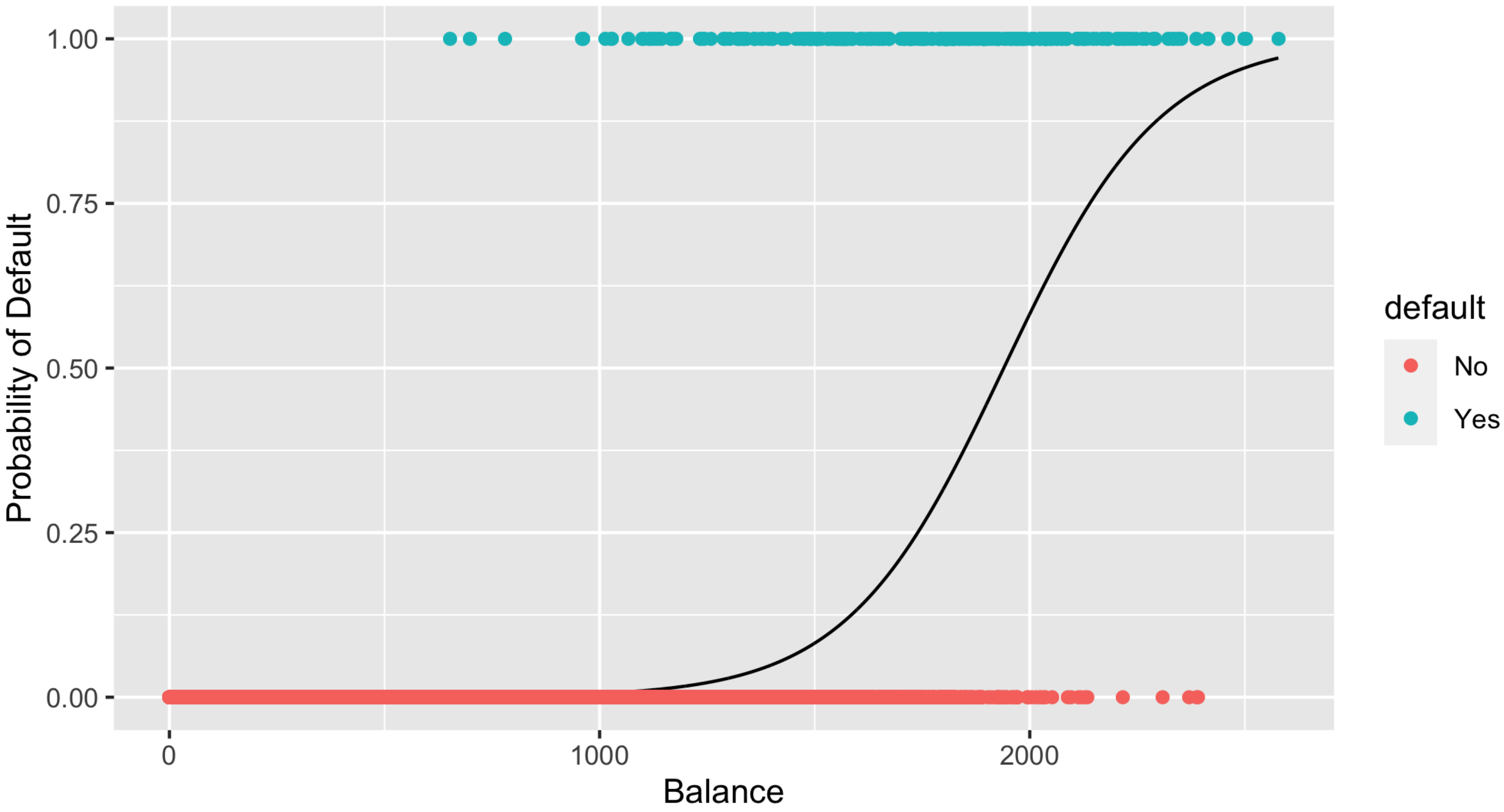
Making predictions in R

- make predictions based on training data

```
lodds <- predict(def_logmod1, type = "link")#log odds

probs <- predict(def_logmod1, type = "response") #probabilities

preds_lodds = ifelse(lodds > 0, "Yes", "No") #using log odds

preds_probs = ifelse(probs > 0.5, "Yes", "No") #using probabilities

all(preds_lodds == preds_probs) #prove to yourself these are the same th

confusionMatrix(as.factor(preds_probs), train$default) #confusion matri
```

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(default_logits)/(1 + exp(default_logits))

probabilities_to_plot <- data.frame("balance" = x_range,
                                    "probabilitiy_of_default" =  default

head(probabilities_to_plot)

ggplot(probabilities_to_plot, aes(x = balance, y = probabilitiy_of_defau
  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") #logi
test_preds_lodds <- ifelse(test_lodds > 0, "Yes", "No") #using logits
confusionMatrix(as.factor(test_preds_lodds), test$default) #confusion ma
```

Multiple Logistic Regression & Confounders

- ISLR 4.3.4
- Combine what you know about
  - Multiple linear regression
  - Logistic regression
- "Results obtained form one predictor may be quite different from those obtained using multiple predictors, especially when predictors are correlated."

Goodness of Fit

- Deviance - the deviance of the fitted logistic model from an ideal model that perfectly fits the data
- AIC - considers the deviance and the number of parameters
- Compare two models using an ANOVA

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, xaringanthemer, remark.js, knitr, R Markdown