MONASH University

ETC3250/5250: Introduction to Machine Learning

Categorical response regression

Lecturer: *Professor Di Cook*

Department of Econometrics and Business Statistics

ETC3250.Clayton-x@monash.edu

Week 3a

Categorical responses

In **classification**, the output Y is a categorical variable. For example,

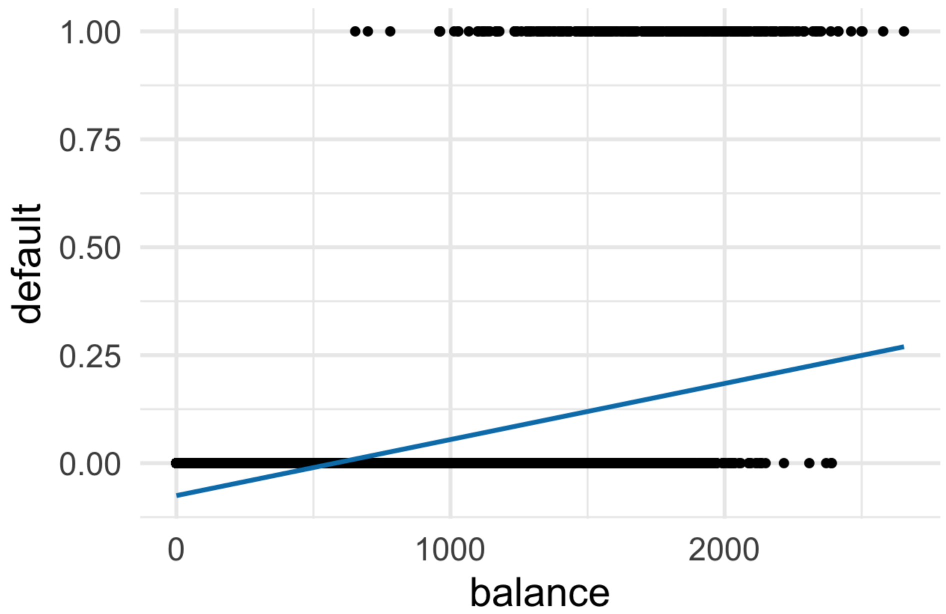
- Loan approval: $Y \in \{\text{successful}, \text{unsuccessful}\}$
- Type of business culture: $Y \in \{\text{clan}, \text{adhocracy}, \text{market}, \text{hierarchical}\}$
- Historical document author: $Y \in \{\text{Austen}, \text{Dickens}, \text{Imitator}\}$
- Email: $Y \in \{\text{spam}, \text{ham}\}$

Map the categories to a numeric variable, or possibly a binary matrix.

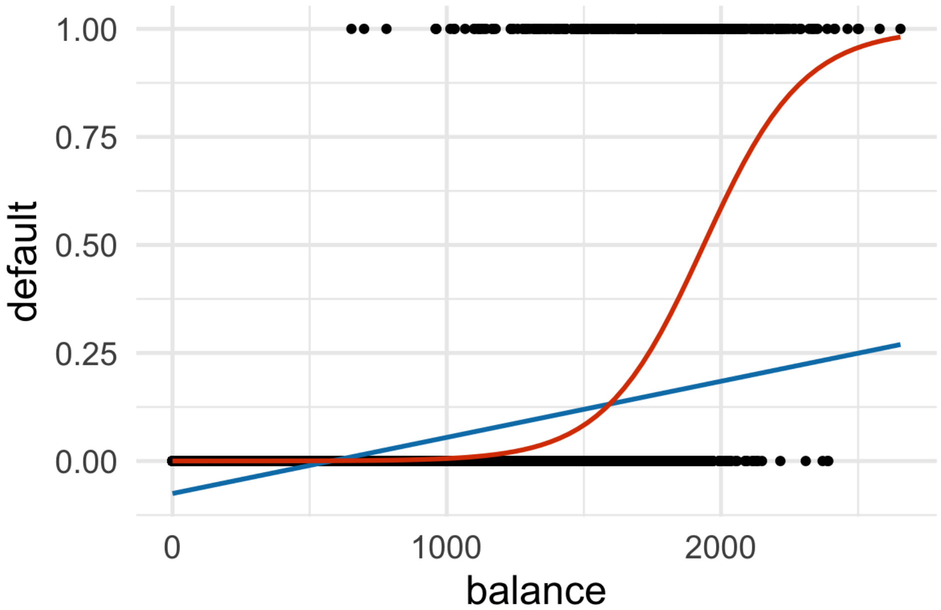
When linear regression is not appropriate

Consider the following data `simcredit` in the ISLR R package (textbook) which looks at the default status based on credit balance.

> **?** Why is a linear model not appropriate for this data?



Modelling binary responses



Orange line is a loess smooth of the data. It's much better than the linear fit.

- To model **binary data**, we need to link our **predictors** to our response using a *link function*. Another way to think about it is that we will transform Y, to convert it to a proportion, and then build the linear model on the transformed response.
- There are many different types of link functions we could use, but for a binary response we typically use the logistic link function.

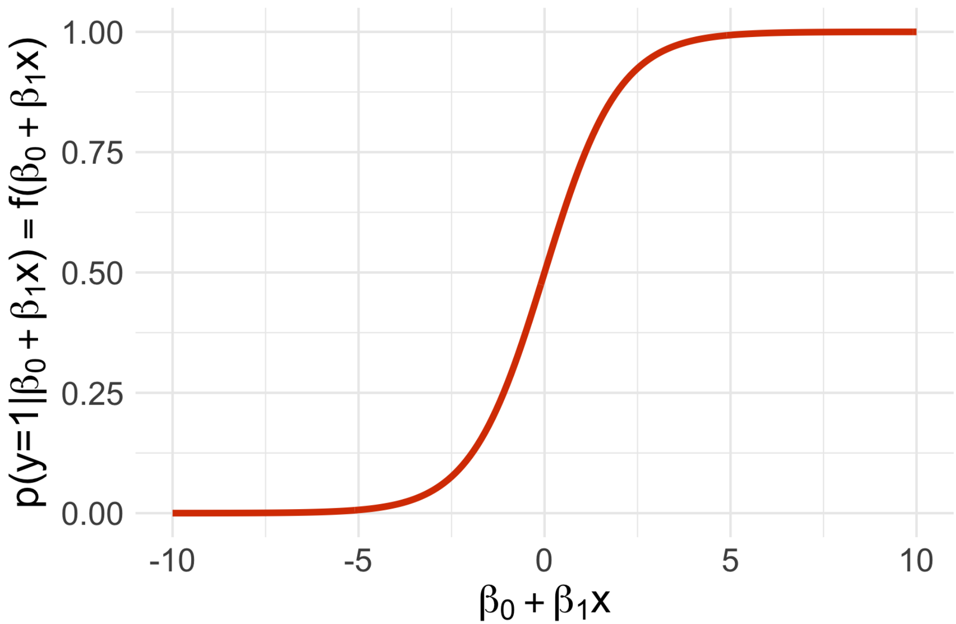
The logistic function

Instead of predicting the outcome directly, we instead predict the probability of being class 1, given the (linear combination of) predictors, using the logistic link function.

$$p(y = 1|\beta_0 + \beta_1 x) = f(\beta_0 + \beta_1 x)$$

where

$$f(\beta_0 + \beta_1 x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}$$



Logistic function

Transform the function:

$$y = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}$$

$$\longrightarrow y = \frac{1}{1/e^{\beta_0+\beta_1 x}+1}$$

$$\longrightarrow 1/y = 1/e^{\beta_0+\beta_1 x} + 1$$

$$\longrightarrow 1/y - 1 = 1/e^{\beta_0+\beta_1 x}$$

$$\longrightarrow \frac{1}{1/y-1} = e^{\beta_0+\beta_1 x}$$

$$\longrightarrow \frac{y}{1-y} = e^{\beta_0+\beta_1 x}$$

$$\longrightarrow \log_e \frac{y}{1-y} = \beta_0 + \beta_1 x$$

Transforming the response $\log_e \frac{y}{1-y}$ makes it possible to use a linear model fit.

The left-hand side, $\log_e \frac{y}{1-y}$, is known as the log-odds ratio or logit. 🎲

The logistic regression model

The fitted model, where $P(Y = 0|X) = 1 - P(Y = 1|X)$, is then written as:

$$\log_e \frac{P(Y=1|X)}{1-P(Y=1|X)} = \beta_0 + \beta_1 X$$

Multiple categories: This formula can be extended to more than binary response variables. Writing the equation is not simple, but follows from the above, extending it to provide probabilities for each level/category. The sum of all probabilities is 1.

Interpretation

- **Linear regression**
  - β_1 gives the average change in Y associated with a one-unit increase in X
- **Logistic regression**
  - Increasing X by one unit changes the log odds by β_1, or equivalently it multiplies the odds by e^{β_1}
  - However, because the model is not linear in X, β_1 does not correspond to the change in response associated with a one-unit increase in X

Maximum Likelihood Estimation

Given the logistic $p(x_i) = \frac{1}{e^{-(\beta_0+\beta_1 x_i)}+1}$

We choose parameters β_0, β_1 to maximize the likelihood of the data given the model. The likelihood function is

$$l_n(\beta_0, \beta_1) = \prod_{y_i=1,i}^{n} p(x_i) \prod_{y_i=0,i}^{n} (1 - p(x_i)).$$

It is more convenient to maximize the *log-likelihood*:

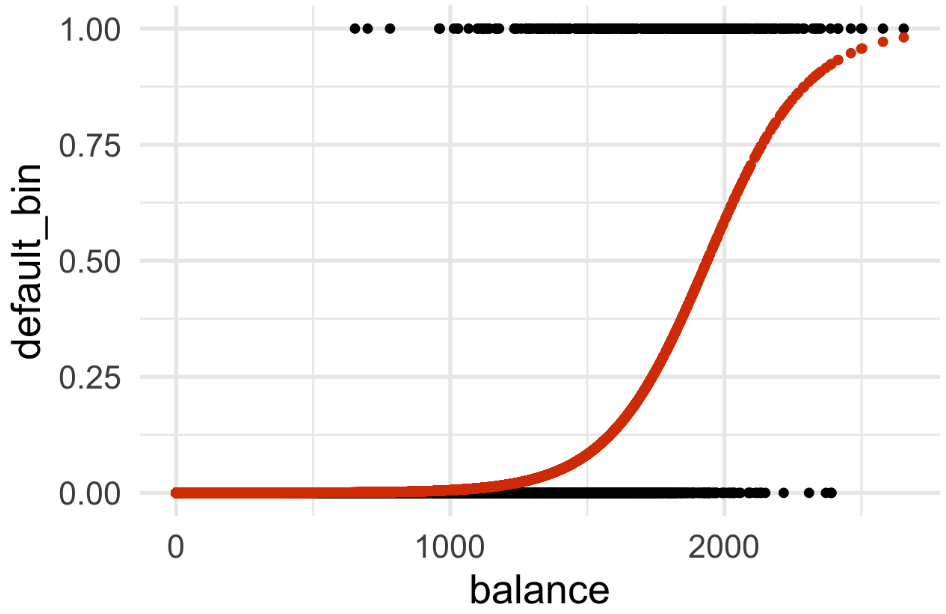
$$\max_{\beta_0,\beta_1} \; \log l_n(\beta_0, \beta_1) = \max_{\beta_0,\beta_1} \; -\sum_{i=1}^{n} \log \left(1 + e^{-(\beta_0+\beta_1 x_i)}\right)$$

Making predictions

With estimates from the model fit, $\hat{\beta}_0, \hat{\beta}_1$, we can predict the **probability of belonging to class 1** using:

$$p(y = 1|\hat{\beta}_0 + \hat{\beta}_1 x) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 x}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 x}}$$

Round to 0 or 1 for class prediction.



Orange points are fitted values, $\hat{y}_i$. Black points are observed response, y_i. (Residual is the difference between observed and predicted.) To generate categorical predictions, round the fitted values.

Fitting credit data in R

We use the `glm` function in R to fit a logistic regression model. The `glm` function can support many response types, so we specify `family="binomial"` to let R know that our response is *binary*.

```
logistic_mod <- logistic_reg() %>%
  set_engine("glm") %>%
  set_mode("classification") %>%
  translate()

logistic_fit <-
  logistic_mod %>%
  fit(default ~ balance,
      data = simcredit)
```

Examine the fit

```
tidy(logistic_fit)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept) -10.7      0.361       -29.5 3.62e-191
## 2 balance       0.00550  0.000220     25.0 1.98e-137
```

```
glance(logistic_fit)
```

```
## # A tibble: 1 x 8
##   null.deviance df.null logLik   AIC   BIC deviance df.residual  nobs
##           <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int> <int>
## 1         2921.    9999  -798. 1600. 1615.    1596.        9998 10000
```

Write out the model

$\hat{\beta}_0 =$ -10.6513306

$\hat{\beta}_1 =$ 0.0054989

Model fit summary

Null model deviance 2920.6 (think of this as TSS)

Model deviance 1596.5 (think of this as RSS)

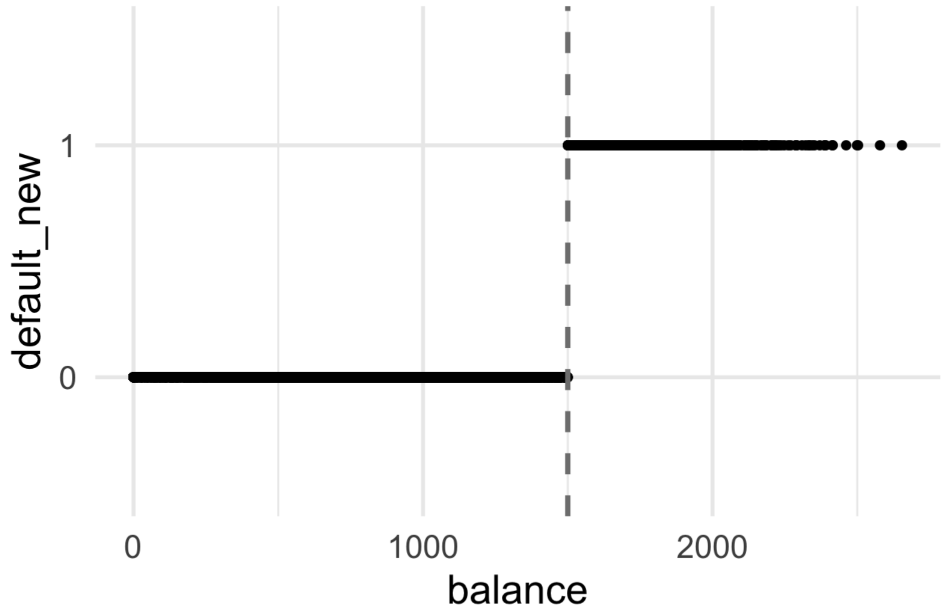
Check model fit

```
simcredit_fit <- augment(logistic_fit, simcredit)
simcredit_fit %>%
  count(default, .pred_class) %>%
  pivot_wider(names_from = "default", values_from = n)

## # A tibble: 2 x 3
##   .pred_class    No   Yes
##   <fct>       <int> <int>
## 1 No           9625   233
## 2 Yes            42   100
```

Note: Residuals not typically useful.

A warning for using GLMs!



Logistic regression model fitting fails when the data is *perfectly* separated.

MLE fit will try and fit a step-wise function to this graph, pushing coefficients sizes towards infinity and produce large standard errors.

Pay attention to warnings!

```
logistic_fit <-
  logistic_mod %>%
  fit(default_new ~ balance,
      data = simcredit)

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

More on supervised classification to come

Logistic regression is a technique for supervised classification. We'll see a lot more techniques: linear discriminant analysis, trees, forests, support vector machines, neural networks.

Lecturer: *Professor Di Cook*

Department of Econometrics and Business Statistics

ETC3250.Clayton-x@monash.edu

Week 3a